NASA/CR-2002-211652


# A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS)

*Troy David Altus*
*George Washington University, Joint Institute for Advancement of Flight Sciences*
*Langley Research Center, Hampton, Virginia*

May 2002

## The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- Email your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/CR-2002-211652

# A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS)

*Troy David Altus*
*George Washington University, Joint Institute for Advancement of Flight Sciences*
*Langley Research Center, Hampton, Virginia*

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Cooperative Agreementt NCC1-387

May 2002

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

## Abstract

BLISS (Bi-Level Integrated System Synthesis) is a decomposition optimization method for engineering systems. The method is characterized by the separate optimization of a relatively few system-level variables and the optimization of potentially numerous local variables. Subsystem optimizations are autonomous and may be conducted concurrently (i.e. on a multiple processor computer). In previous versions of BLISS, optimum sensitivity analysis and system sensitivity data were used to link the subsystem optimization data to the system optimization. The current work replaces both the optimum sensitivity analysis and the system sensitivity equations by the quadratic response surface representations using subsystem optimization results.

The response surface methodology for BLISS achieves the desired improvements while retaining key attributes of previous versions of BLISS: the autonomy of the black box optimizations and the clear separation of the system variables from the potentially numerous local variables. The response surface formulation of BLISS was successfully demonstrated on a simplified conceptual design of a supersonic business jet.

In addition to changes in the overall optimization methods of BLISS, subsystem fidelity was enhanced, accompanied by the necessary modifications to the data flow between subsystem analyses. Documentation of these modifications that have not as yet been tested is included as a reference for future research.

## Acknowledgements

The author would like to thank the following people, without whom this research could not have been possible: Dr. Jaroslaw Sobieski, NASA Langley Research Center, and Prof. Robert Sandusky, George Washington University, for their guidance throughout the course of this project; Lt. Matthew Phillips, U.S. Air Force, for his concurrent BLISS research efforts; Dr. Robert Biedron, NASA Langley Research Center, for his help in CFD implementation; Dr. Jamshid Samareh, NASA Langley Research Center, for assisting in grid-morphing software (MASSOUD); Norma Bean and Lee Kania, GEOLAB at NASA Langley Research Center, for their help in baseline grid generation; G. E. Plassman, Computer Sciences Corporation, for aiding the validation of response surfaces; Michael Park, NASA Langley Research Center, for his programming advice; and Paul E. Escalera, George Washington University, for his help in producing printed versions of this document.

# Table of Contents

# List of Figures and Tables

# Nomenclature

$AR_{HT}$ – tail aspect ratio

$AR_W$ – wing aspect ratio

$\beta$ - Regression coefficient

BB – black box

CCD – central composite design

D – drag, lb

dA – vertical displacement of wing tip leading edge, in

dB – vertical displacement of wing tip trailing edge, in

DOE – Design of Experiments

ESF – engine scale factor

h – altitude, ft

L – lift, lb

L/D – lift to drag ratio

$L_{HT}$ – horizontal tail location, % mean aerodynamic chord (% MAC)

$L_W$ – wing location, % MAC

M – Mach number

R – range, NM

RS – response surface

RSM – response surface methodology

SFC – specific fuel consumption

$S_{HT}$ – horizontal tail surface area, $ft^2$

$S_{REF}$ – wing surface area, $ft^2$

T – throttle setting, %

$T_J$ – Jig Twist (deg)

t/c – thickness to chord ratio

$t_i$ – wingbox sandwich face sheets thicknesses, in

$t_{s,i}$ – wingbox sandwich caliper thicknesses, in

$X_i$ – design variables local to $BB_i$

XL, XU – lower and upper bounds on X, side-constraints

w – weighting factors

$W_E$ – engine weight, lb

$W_F$ – fuel weight, lb

$W_T$ – total weight, lb

Y* – Input to a BB

Y^ – output from a BB

Z – system-level design variables

$\lambda$ – taper ratio

$\Lambda_{HT}$ – horizontal tail sweep, deg

$\Lambda_W$ – wing sweep, deg

$\Theta$ – equivalent reduction in effective lift area due to twist, $ft^2$

$\phi_W$ – Wing incidence angle, deg

$\phi_{HT}$ – Horizontal tail incidence angle, deg

# 1. Introduction

The design of a complex engineering system (e.g. an aircraft) involving a large number of subsystems, modules, or black boxes (BB's) inherently involves a large number of design variables and constraints. The optimization problem can quickly become too large to manage efficiently, and the solution process can become extremely expensive in a computational sense [1]. Decomposition of the problem into more manageable subtasks enables an efficient distribution of work across disciplines. In such decomposition, the design variables and constraints local to a particular module are separated from those that affect the system as a whole. The separation and distribution of work promotes a diverse grouping of human and computer resources, thus conforming to current trends in parallel processing technologies and concurrent engineering.

The original formulation of Bi-Level Integrated System Synthesis, BLISS (referred to as BLISS-98), introduced in [2] and documented in detail in [3], involved the conceptual design of a business jet, optimized for maximum range. During this design analysis, simple empirical and analytical calculations were performed for each of the BB analyses. In [4], the test case was the same supersonic business jet but the fidelity of the analyses was improved. Both previous BLISS formulations share the distinguishing feature that the system objective was range while that of local optimization was a composite function made up of the sum of BB outputs, each weighted by the system sensitivity derivatives for that term. Therefore, the optimum sensitivity derivatives are used as the coupling mechanism between system and local optimizations.

The current version of BLISS (called BLISS-RS herein to denote response surface methodology) is conceptually similar to previous versions. However, in BLISS-RS the

coupling between system and local optimization occurs via quadratic response surfaces instead of an optimum sensitivity analysis. The new framework of BLISS was successfully tested using BB's inherited from previous versions of BLISS.

## 2. BLISS-98 System Architecture

Regardless of the particular decomposition approach, the introduction of the BLISS algorithm begins with formulation of the problem without decomposition. A modular system like BLISS typically optimizes three different types of design variables. First, the system level variables Z affect at least two of the BB's. Secondly, the local variables X are specific to a particular BB. Finally, the coupling variables Y^ are output from a BB, while Y* are coupling variables input to a BB. A statement combining analysis and optimization of a generic system can be written as:

| | | |
|---|---|---|
| Find: | V | **Equation 1** |
| | Where V represents the collection of all design variables in the space {Z \| X \| Y* \| Y^} | |
| Minimize: | F(V) | |
| Satisfy: | $\{g(V)\} \leq 0$, for each BB | |
| | $\{h(V)\} = 0$ | |
| | $\{c(V)\} = Y^* - Y^\wedge = 0$ | |
| | $\{VL \leq X \leq VU\}$. | |

In equation 1, the inequalities {g} represent the behavior constraints local to a BB, and the equalities {h} correspond to the solution of the BB governing equations (BB inner

analysis). The coupling equalitites {c} describe the condition that the output coming from one BB has to match the input to another BB, for a particular coupling variable Y.

Solution of the all-in-one formulation shown above can be very arduous due to the possibly large number of local variables. For this reason, it is beneficial to decompose the problem based on the types of design variables. BLISS is such a decomposition method in that the problem is divided into two steps: local optimization within each BB and the system optimization.

The original formulation of BLISS relied predominately on gradients to guide the search toward an optimum. It depended on system and sensitivity analyses, local optimizations inside the BB's, and the system optimization. Each cycle through the BLISS procedure improved the design in two steps; first by optimizing each BB for the local design variables X while holding Z constant; and next, a system-level optimization that treats Z as the design variables [3],[4]. Figure 1 depicts the BLISS-98 process.

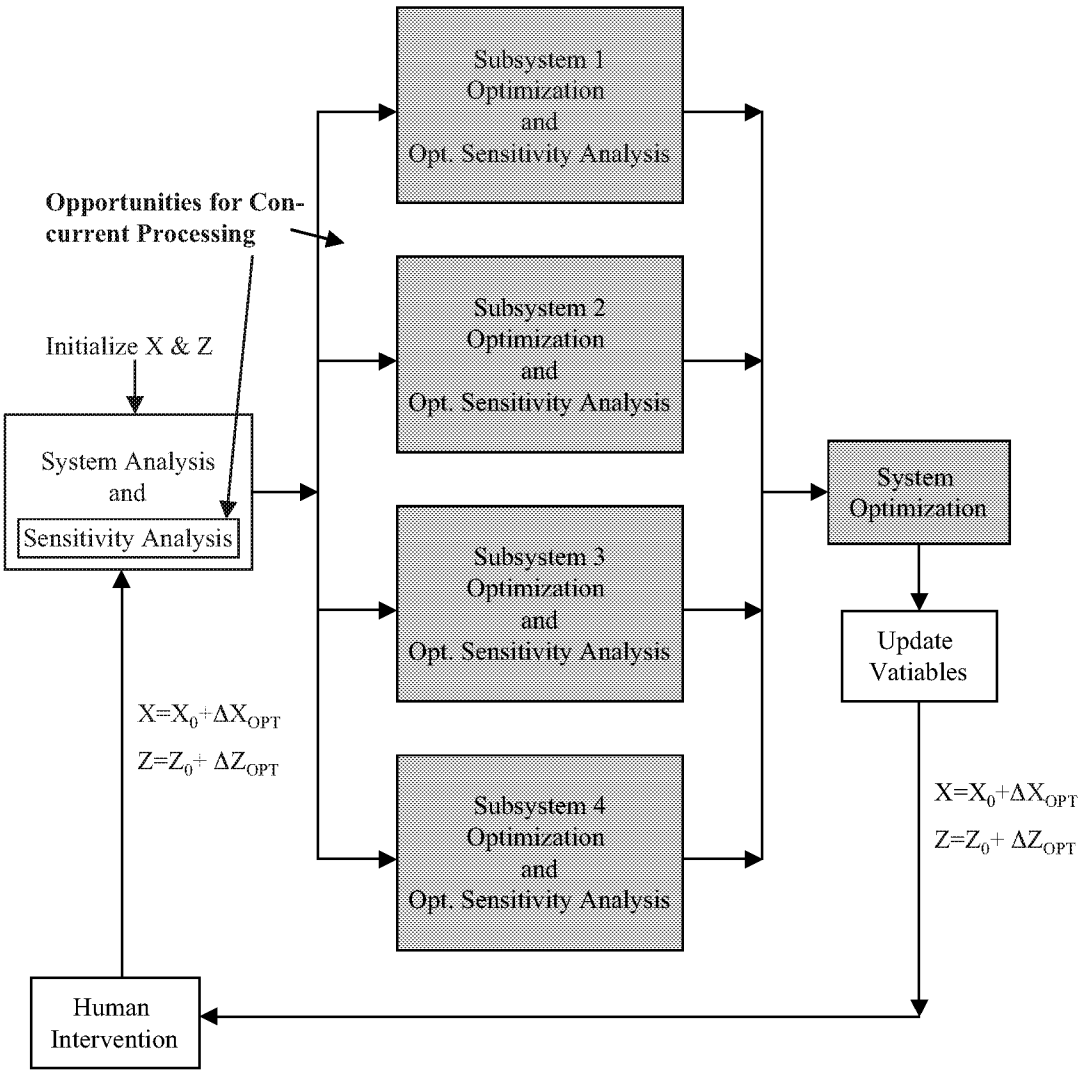


**Figure 1. BLISS-98 Cycle**

After initialization, the BLISS-98 performs a system analysis and sensitivity analysis in which Y and the derivatives of Y with respect to Z and X are computed. A linear approximation to the system objective (an element of Y – namely Range output from the performance BB) as a function of Z and X is established using the above derivatives. That approximation is adopted as the objective function in BB optimizations that follow. In each BB, the Z and Y variables are frozen and an improvement in the objective function is achieved by the local optimizations that use local X separately in

each module. This is followed by computation of the derivatives of the optimum X with respect to the parameters Z and Y.

The second step seeks improvement for the system-level variables Z and is linked to the first step by the derivatives of $X_{OPT}$ with respect to parameters Z and Y. The derivatives are used to extrapolate each subdomain optimum as a function of Z and Y. The functional relation Y=Y(Z) is approximated by extrapolation based on the system sensitivity analysis. These steps alternate until convergence. Note that the output of step 1 is an optimum change in the local design variables, $\Delta X_{OPT}$, in the presence of constant Z, and the output of step 2 is an optimum change in system design variables, $\Delta Z_{OPT}$.

BLISS-98 was successfully implemented for the test case of a design of a supersonic business jet. A detailed description of the results obtained can be found in [4]. However, its dependence on the system analysis and derivative information proved to be computationally costly.

## 2.1. BLISS-98 System Variable Flow

Figure 2 shows the general flow of data between the various modules of BLISS-98. The system variables (those which affect more than one black box directly) are shown in the dashed boxes. Variables output from a BB are specified using (^), and those input to a BB are designated by (*).

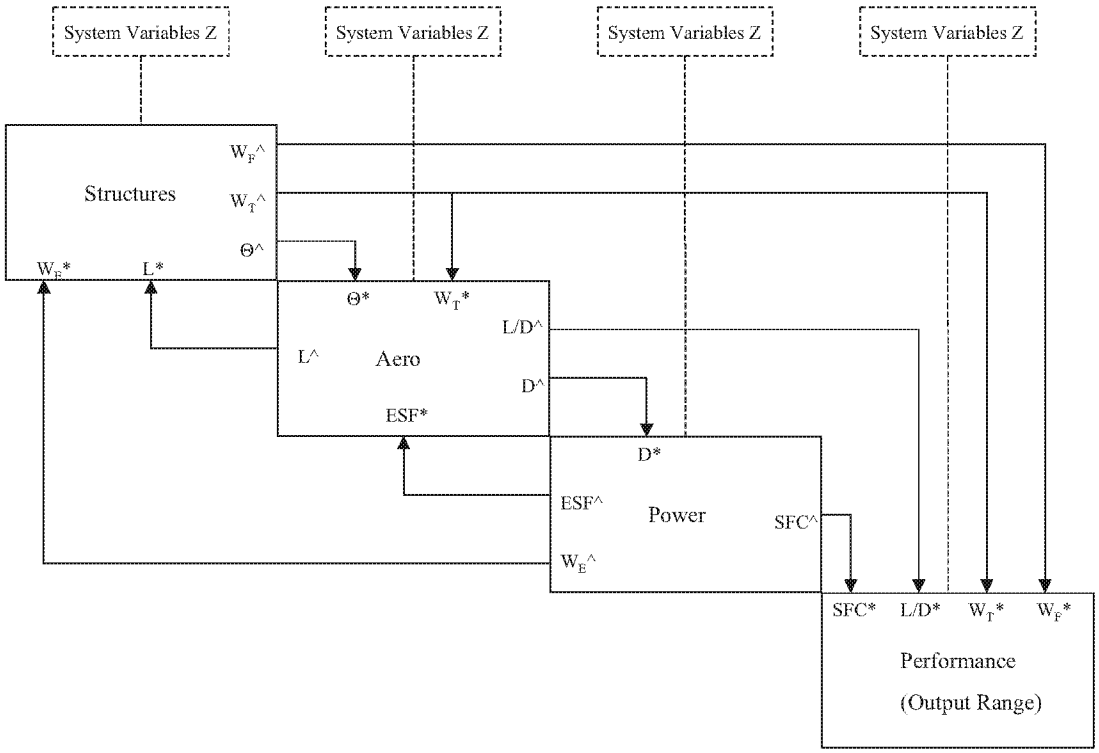


**Figure 2. BLISS-98 System and Coupling Variable Flow**

## 2.2. BLISS-98 Subsystem Analyses

The subanalysis modules included in BLISS-98, as well as the general data flow among these modules, were used as a benchmark for BLISS-RS. Therefore, a brief discussion of the analysis involved in each module is presented here.

### Structures BB

Figure 3 is a schematic representation of the structures black box, showing system variable flow and intermediate operations. The structures BB takes in all Z-level variables excluding atmospheric parameters, along with Lift from the aerodynamics BB and engine weight from the power BB.

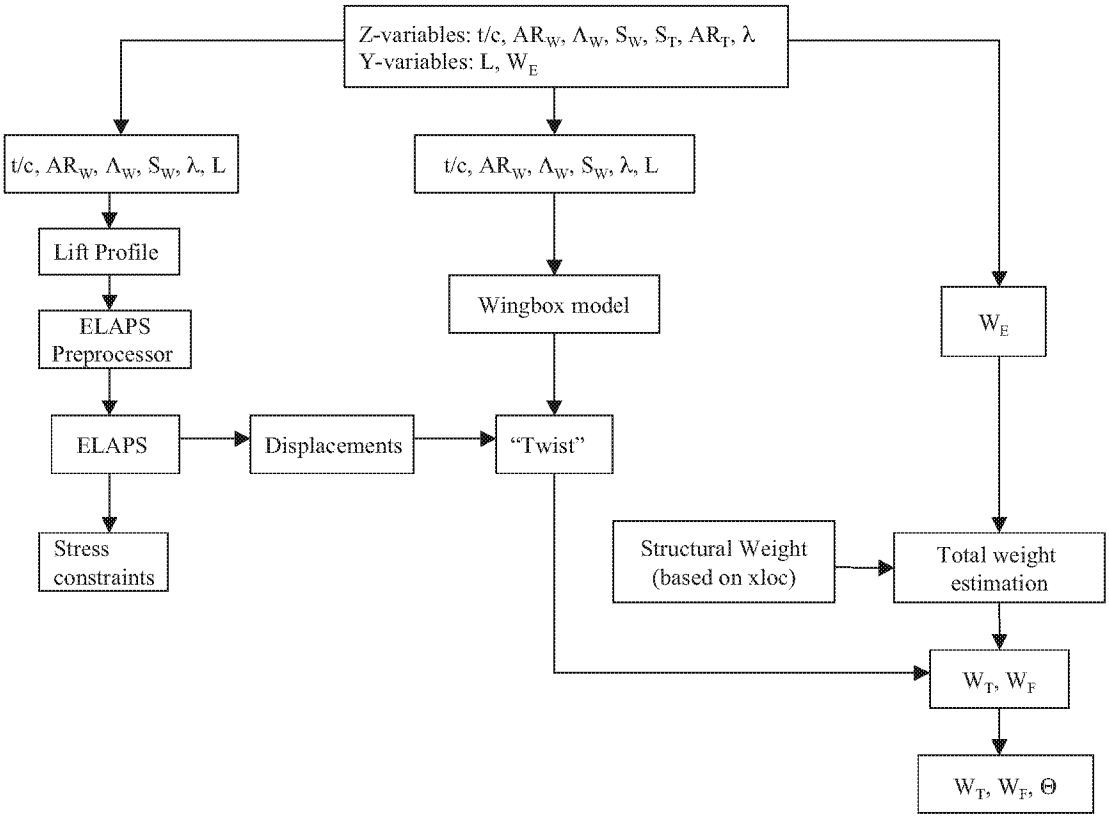


**Figure 3. System Variable Flow for BLISS-98 Structures Black Box**

The aerodynamic loads are generated within the structures module in a pre-processor to generate the appropriate input into the structural analysis package. The lift loads on the wing are determined via the Shrenk approximation: a spanwise average of an elliptical lift distribution and a trapezoidal distribution that reflects the wing chord taper, as shown in Figure 4 [5]. This load distribution is based on half of the total aircraft weight. Lifting effects of the fuselage and tail are ignored.

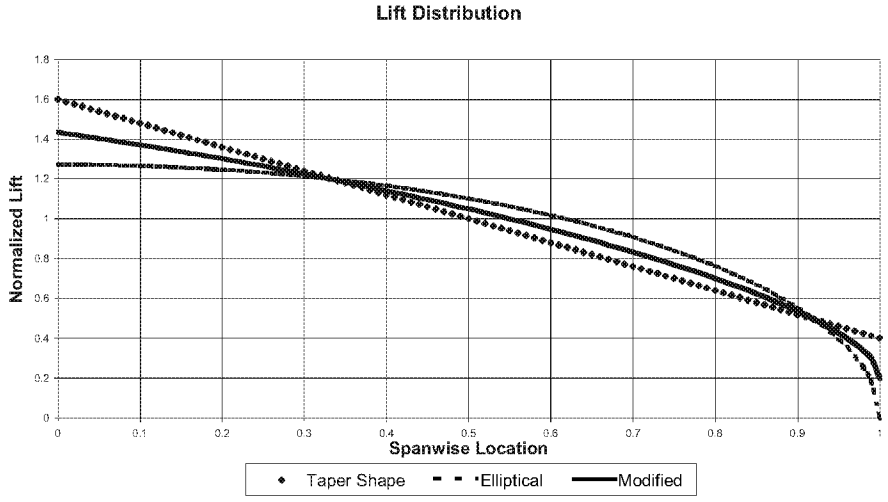


**Figure 4. BLISS 1998 ELAPS Lift Distribution [4]**

The structures BB uses as its primary analysis tool the Fortran-77 based Equivalent Laminated Plate Solution (ELAPS), described in [6-9]. ELAPS analyzes trapezoidal sections, or elements, that represent whole lifting surfaces (plate segments) or fuselages (shell segments). For each segment, a Ritz-based method is used to minimize strain energy, yielding polynomial equations for static deflections and internal stresses.

Although the accuracy of ELAPS has been shown to be slightly below that of finite element codes, many beneficial aspects lend it nicely to multidisciplinary optimization. The main advantage lies in the fact that, compared to finite element codes, ELAPS generates far fewer degrees of freedom, thus reducing the required computational time and expense [10]. The input data is also much simpler and faster to develop. The model can consist of one segment representing an entire simple wing, or thousands of segments of a partitioned wing. Therefore, the adaptability of the model makes ELAPS an attractive option for all phases of the design process.

Within the BLISS framework, stresses are analyzed along a three-segment wingbox. Each wingbox consists of the top and bottom sandwich panels of different

thicknesses and sandwich webs that are identical in the front and rear of the wingbox. The front spar of the wing box is located at 10% of the chord length and the rear spar lies at 70% of the chord length. Figure 5 depicts the configuration of the ELAPS model used by BLISS.

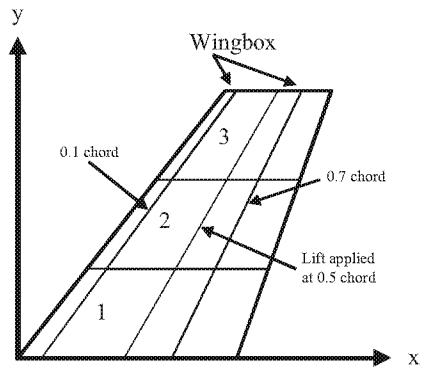


**Figure 5. Wingbox configuration used by ELAPS [4]**

The top and bottom panels as well as the webs have the thickness of the sandwich face sheets ($t_i$) and the sandwich caliper thickness ($t_{s,i}$) as design variables, depicted in Figure 6. ELAPS models such a built-up structure by representing each face and the core as separate elements linked in a common coordinate grid.

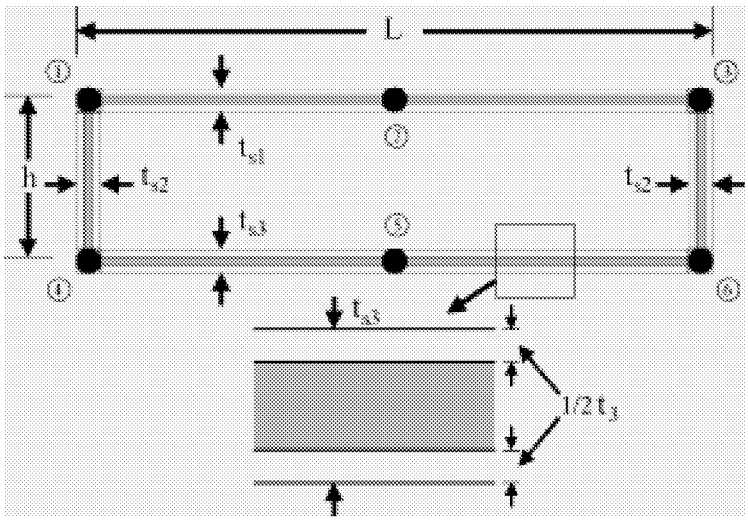


**Figure 6. Wingbox Cross-sectionl [4]**

The stresses calculated by ELAPS are used in formulation of a set of local constraints. The displacements are combined with the wingbox model to yield the value labeled "twist" in Figure 3. "Twist" is formulated as a change in the total lift of the wing divided by dynamic pressure, q, having units of $ft^2$. Thus, it can be viewed as a *change in effective wing area due to changes in the local chord angle of attack induced by the wing twist and bending.* Finally, the total structural weight is calculated using the structures BB local variables. This is combined with the engine weight to estimate the total weight of the aircraft. Fuel weight is estimated by empirical relationships.

**Aerodynamics BB**

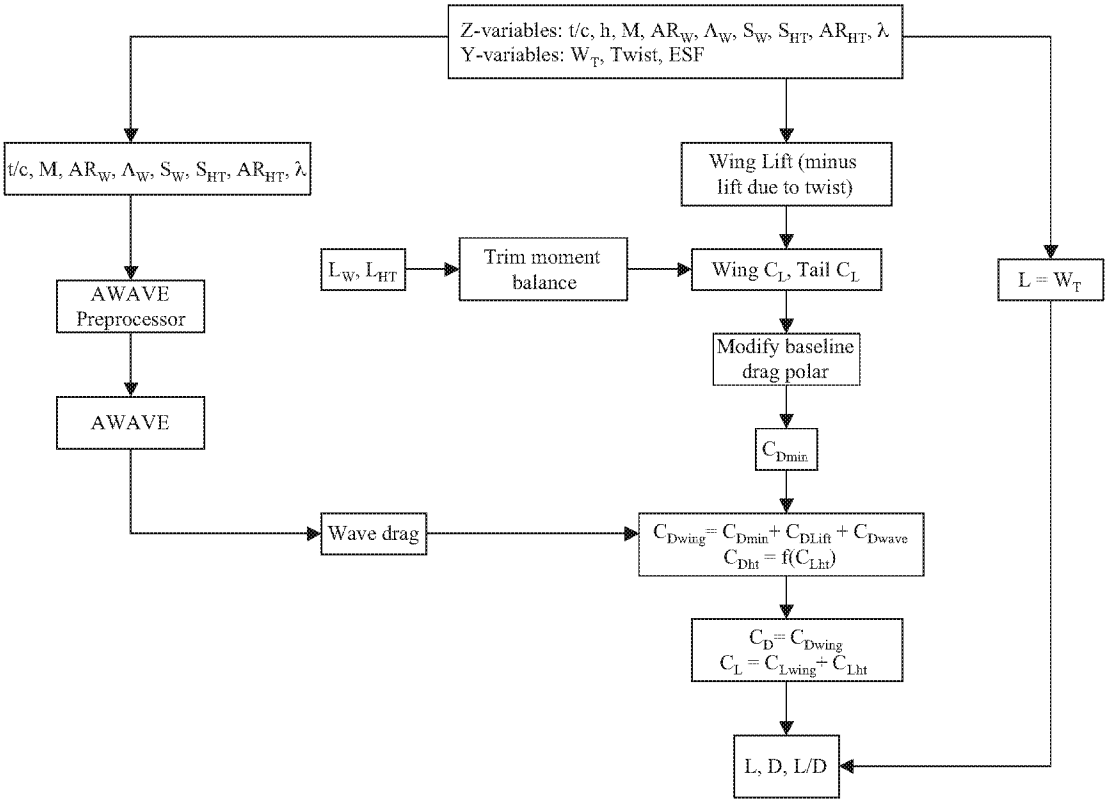


Figure 7 is a schematic representation of the aerodynamics black box, showing system variable flow and intermediate operations. The aerodynamics BB takes in all Z-

level variables, along with total weight and twist from the structures BB, and engine scaling factor from the power BB.

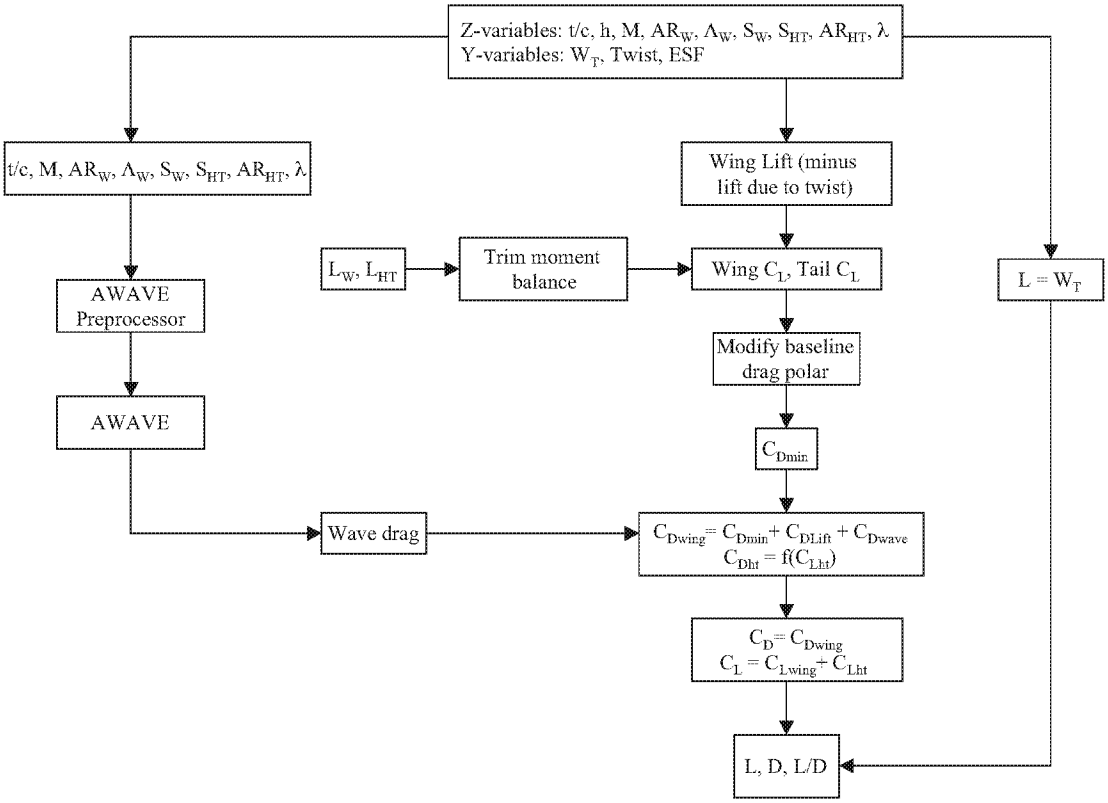


**Figure 7. System Variable Flow for BLISS-98 Aerodynamics Black Box**

The left side of the chart shows the inputs used by the AWAVE preprocessor. AWAVE is a simplified version of the Harris far-field wave drag program. The AWAVE Fortran code computes wave drag on the basis of the aircraft's cross-sectional distribution along the centerline, hence it requires data about the entire configuration geometry to enable the area ruling [4].

A baseline drag polar is modified using ESF to account for wave drag of the engine nacelle. This is combined with the wave drag from AWAVE and drag due to lift to yield the wing's drag coefficient $C_{Dwing}$. The drag coefficient for the horizontal tail, $C_{Dht}$, is calculated from an empirical relation with $C_{Dwing}$. This leads to the value of lift to

drag ratio, L/D. Since BLISS models only the cruise flight regime, the total weight input set equal to total lift, L. Total drag is easily found dividing total lift by L/D.

## Power BB

The power black box is a relatively simple in form. The drag input from the aerodynamics BB is set equal to thrust for the cruise condition. Engine weight is found through an empirical formula involving drag and ESF. SFC is determined using an engine deck approximation model (quadratic) that takes in values for thrust and throttle setting.

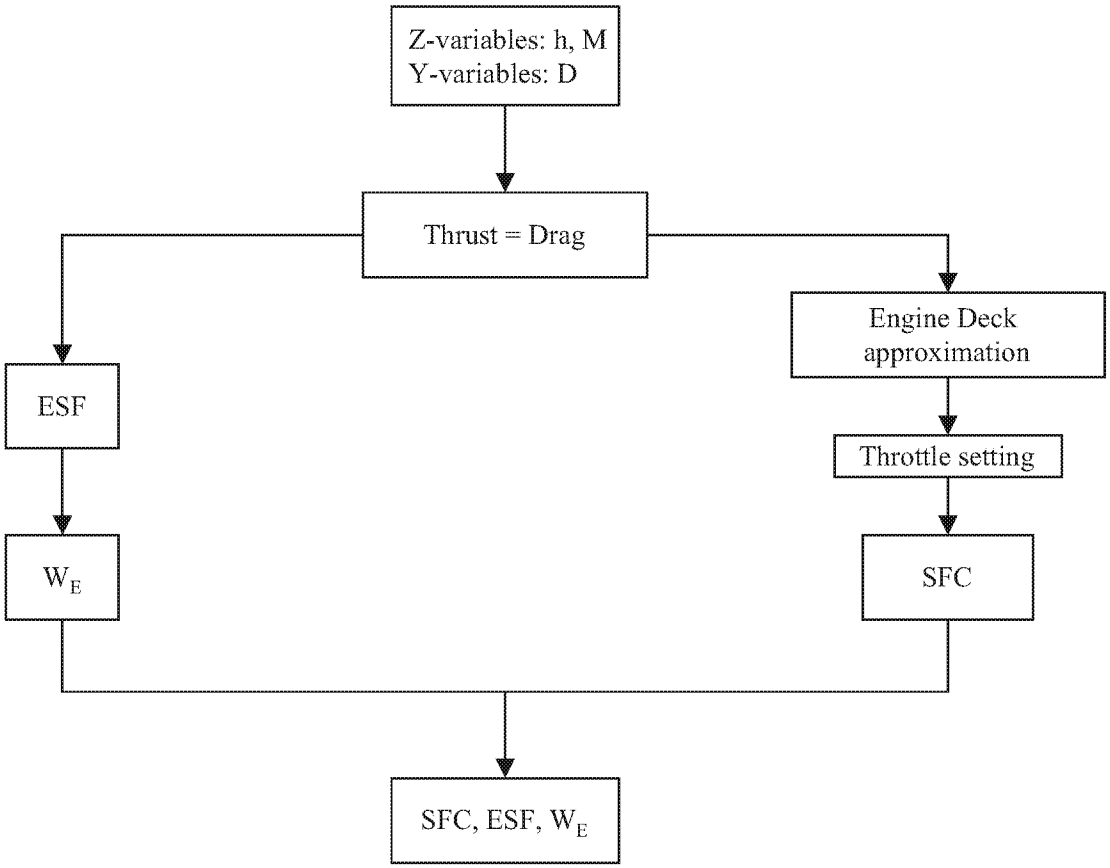


**Figure 8. System Variable Flow for BLISS-98 Power Black Box**

### Performance BB

The simplest black box is the performance, or Range BB. It is made up entirely of the Breguet Range equation.

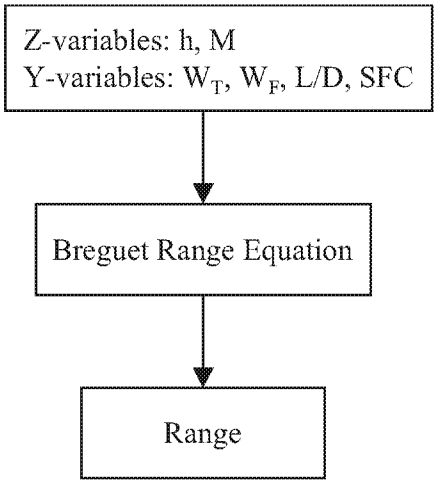


**Figure 9. System Variable Flow for BLISS-98 Range Black Box**

## 3. BLISS-RS

The motivation for the current research is a direct consequence of the aforementioned shortcomings of BLISS-98. At the same time, the primary attributes of BLISS (clear separation of system and local design variables and the autonomy of BB's) need to be preserved. Reformulating BLISS to incorporate response surfaces is a logical choice for addressing these problems. Since RS methods efficiently explore the entire design space, the optimization procedure is less likely to terminate at a local optimum. Also, the expense associated with calculating the local and system sensitivity derivatives is eliminated in the present method. However, this reduction in computational labor is partially offset by the labor to generate a number of response surfaces for each BB [11].

Figure 10 shows the general BLISS-RS procedure. The first operation involves initialization of the system level variables Z, local variables X, and weighting factors w. Next, the coupling variables are initialized, either through a system analysis or an educated guess. After all variables are initialized, initial upper and lower bounds ($LB_0$ and $UB_0$) are selected for all variables. A DOE pattern is then used to create a dispersion of inputs to each BB. The BB's are optimized locally for each input, and then response surfaces are fitted through each optimal BB output. This collection of response surfaces is used for system optimization, allowing for rapid extraction of coupling data. If the system successfully converges, a final analysis is performed to retrieve local variables X. If convergence has not been achieved, the response surfaces are adjusted, and the system optimization process is repeated.

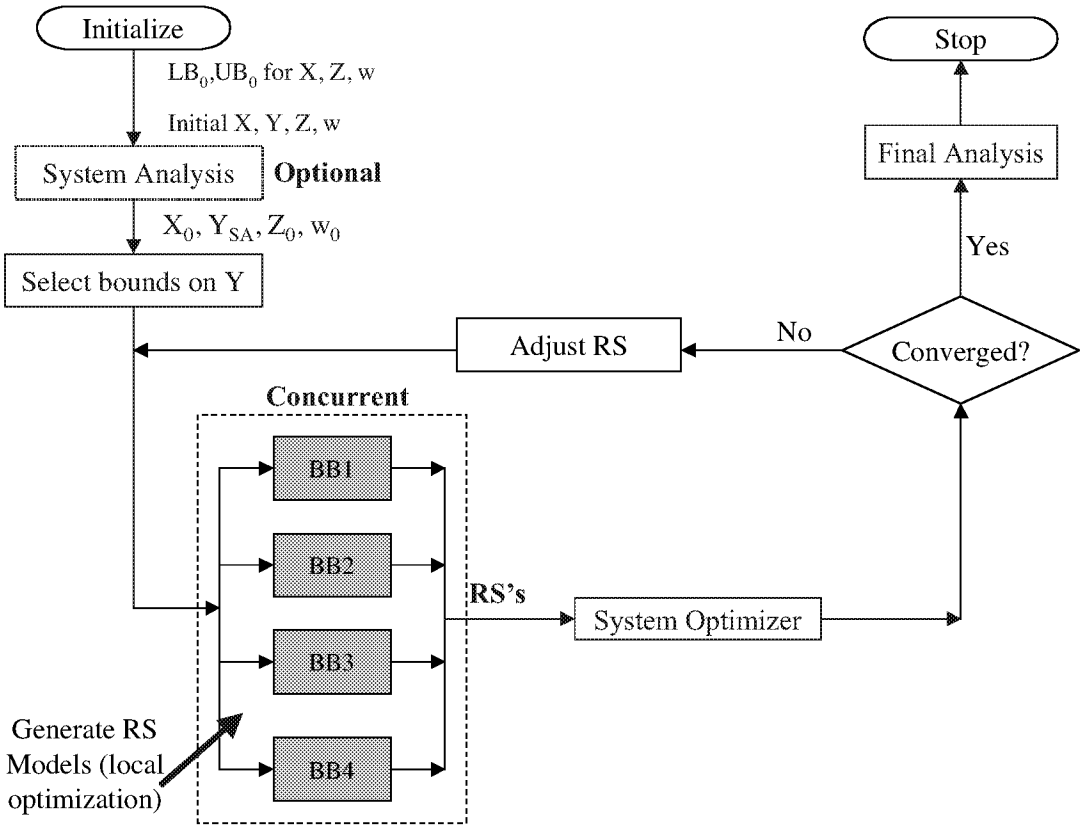


**Figure 10. BLISS-RS Flowchart**

## 3.1. Local Optimization

Formulation of a local BB objective function can be accomplished by observing the data flow between the various modules. In Figure 2, the system objective is the range, which is output from the performance BB. The range is directly related to those quantities entering the BB - namely total weight, fuel weight, specific fuel consumption, and lift-to-drag ratio. To illustrate an indirect influence of a BB output to the range, consider the wing twist. The twist information is sent to the aerodynamics module, where it is used to modify the wing shape and thus alter the lift and drag that the aerodynamics BB sends to the performance BB.

Naturally, the structures BB should be optimized such that the range is increased as much as possible. However, some of the outputs of the structures BB influence performance directly, while others have an indirect effect. Therefore, it is not clear how much relative importance needs to be given to each output.

This problem is addressed by assigning weight factors w to each of the outputs of a BB, then optimizing for a composite objective function. These weight factors are analogous to the system sensitivity derivatives used in previous versions of BLISS. The composite objective function is a weighted sum of the contributions of each output. For example, the composite objective for the structures BB would be the sum of total weight, fuel weight, and wing twist, each multiplied by a weight factor. Since the actual values of w are unknown, a RS is formed in the space of {Z | Y* | w} and the task of determining the values of w is now left up the system optimizer.

Given vectors of system variables, coupling variables, and weight factors, a black box can be optimized for a composite objective function, F with the satisfaction of all

local constraints. The result is a point F in the space {Z | Y* | w}. It represents the best contribution to the "synthetic" cost function the BB can make given Z, Y, and local constraints. The term "synthetic" is used because the system objective is not considered at this point; instead, the function that is minimized is made up of multiple terms that represent all outputs of the BB. Each of these terms includes a weighting factor w that dictates the relative importance of each response. The formation of the BB response surfaces (i.e. the BB optimizations) can be stated formally for BB as:

| | | |
|---|---|---|
| Given: | Z, Y*, $w$ | **Equation 2** |
| Find: | X local to BB | |
| Minimize: | $F_k = \sum w_j Y^{\wedge}_i$ | |
| Satisfy: | {h} = 0 | |
| | {g} ≤ 0 | |
| | {XL ≤ X ≤ XU} | |

Where $F_k$ is the BB contribution to the system objective. The constraints {h} = 0 in equation 2 represent the analyses (ELAPS, for example). The {g} constraints could stand for physical limitations (material properties, dimensional tolerances, etc.).

Using a DOE-based point placement method, Z, Y*, and w are varied to yield a locally optimized point in the space {Z | Y* | w} subject to local constraints. Many such points are generated to produce a cloud of points through which a quadratic response surface can be fit. A "surrogate" of each BB is obtained, and this approximation of each BB analysis can then be used in a system-level optimization.

## 3.2. System Optimization

Having obtained the quadratic representations of each BB, the next BLISS action involves reading values from the quadratic RS's to achieve an improvement in the system objective while satisfying the coupling constraints, {c}=0. The coupling constraints are easily recognized by the fact that the output $BB_i$ must be equal to the corresponding input to $BB_j$. For example, the total lift generated in the aero BB must equal that passed along as an input to the structures BB. During system optimization, a coupling variable Y^ is not directly sent from $BB_i$ as an input to $BB_j$. Rather, the system optimizer proposes an input to $BB_j$, Y*, and a constraint is introduced in the form Y^ - Y* = 0. In Figure 11, the coupling constraints are shown graphically as circles where data flow channels meet.

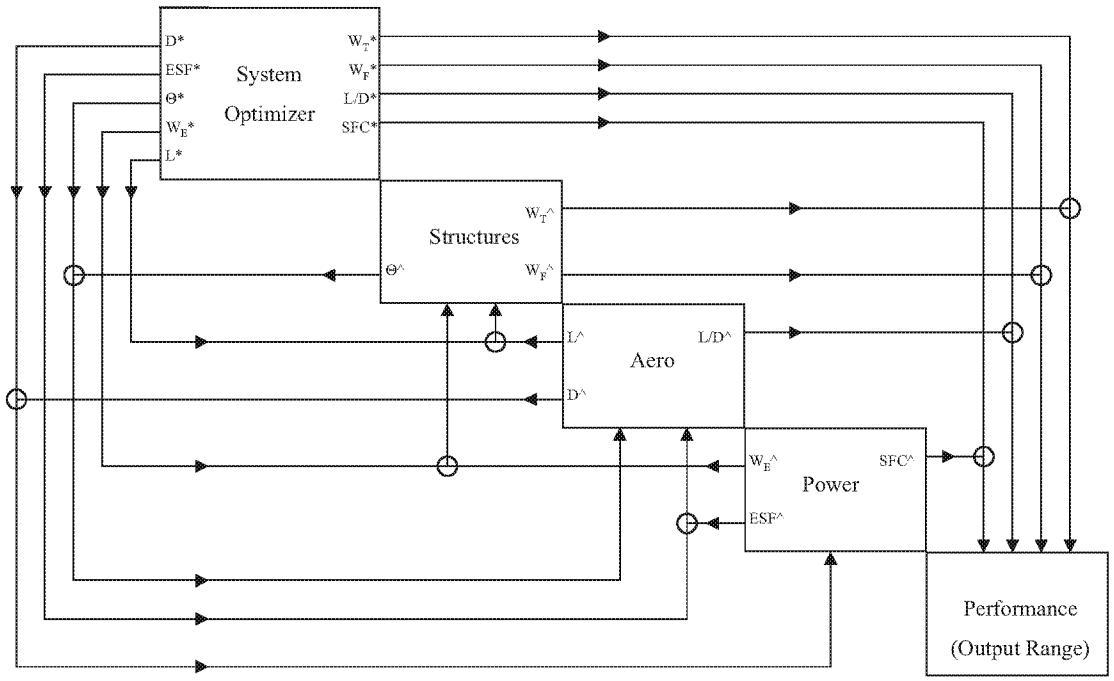


**Figure 11. BLISS-RS System Optimizer**

The system optimization can be formally stated as:

| | | |
|---|---|---|
| Given: | $\beta_j$ for each BB | **Equation 3** |
| Find: | Z,Y*, *w* | |
| Maximize: | $F = Range$ | |
| Satisfy: | $\{c\} = 0$ | |
| | $\{g\} \leq 0$ | |
| | $\{XL \leq X \leq XU\}$ | |

In the above expression, $\beta_j$ are the regression coefficients for the response surface describing the $j^{th}$ output of a particular BB.

## 4. Response Surface Methodology

The underlying goal of many types of experimentation is to correlate an output response to a set of factors of interest to the researcher. These relationships can be attained by constructing a model that describes the response over various values of the factors of interest. These *response surfaces* can be generated as a graphical means of displaying such relationships. Response surfaces can be analyzed to determine optimum combinations of input factors, or to explore relevant tradeoffs when multiple responses are involved. Figure 12 demonstrates such features.

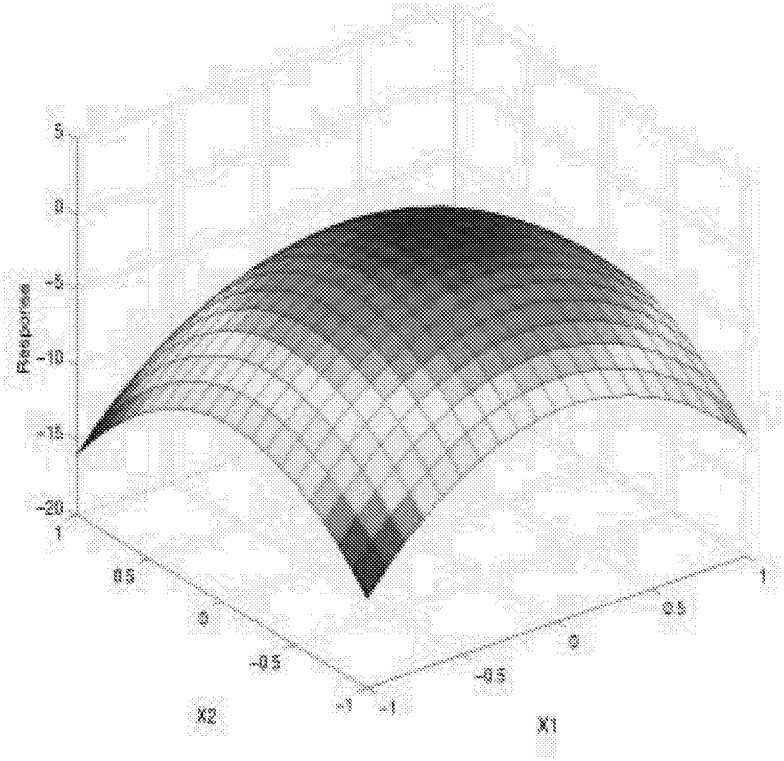


**Figure 12. Generic Example of a Two-input Response Surface**

With regard to design of a complex engineering system, the response surface methodology (RSM) allows the design space to be efficiently explored to determine the values of the design variables that optimize performance characteristics subject to system constraints [12]. RSM is used to obtain the mathematical models that approximate the functional relationships between performance characteristics and design variables [13]. Various design of experiments techniques, such as the central composite design (CCD) and the Box-Behnken design, are used to sample the design space efficiently [14]. With these approaches, experiments are performed at statistically selected locations in the design space. The resulting data is then used to construct response surface models through least squares regression.

However, these, like most other experimental designs were formulated with physical experiments in mind where the measurement variance of the response is the main concern. This stems from the fact that the output response of a physical system usually exhibits some degree of variability with the experiment repeated using the same inputs. Conversely, computer experiments (commonly used in engineering design) are deterministic, that is to say that there is no measurement error and no variability in response outputs given multiple runs of the same test using the same inputs [13]. Therefore, experimental designs constructed to minimize variability of measurements may not be the best choice for computer experiments.

## 4.1. Construction of Empirical Models

### 4.1.1. Linear Regression

In most applications of RSM, it is necessary to develop an approximation model to the true response surface. Approximation is necessary since, for most cases, the underlying function that drives the response is an unknown physical mechanism [12]. Multiple regression is used to generate an empirical model.

A multiple linear regression model with $n$ independent variables takes the form

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \varepsilon \qquad \textbf{Equation 4}$$

where $y$ represents the response value, $x_i$ are the independent variables (*predictor variables* or *regressors*), $\beta_i$ are unknown partial regression coefficients, and ε is the model error. This is a linear model because it is a linear function of the unknown

parameters $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$. This model describes a plane for two independent variables and a hyperplane in higher dimensions for three or more independent variables.

Aany regression model that is a linear function of the regression coefficients is a linear regression model, regardless of the shape of the surface that it generates [12]. For example, interaction terms could be added to equation 3 ($n$=2) to give

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \varepsilon. \qquad \textbf{Equation 5}$$

But if we let $x_3 = x_1 x_2$ and $\beta_3 = \beta_{12}$, then equation 5 becomes a standard multiple linear regression model. Higher order models can be generated using similar techniques.

The method of least squares is typically employed to evaluate the $\beta$'s in equation 5. If the response is observed $m$ times, $y_1, y_2, \ldots, y_m$, where $m>n$, equation 5 can be rewritten as

$$y_i = \beta_0 + \sum_{j=1}^{n} \beta_j x_{ij} + \varepsilon_i, \qquad i=1,2,\ldots,m \qquad \textbf{Equation 6}$$

In matrix notation, this can be written as

$$\{y\} = [X]\{\beta\} + \{\varepsilon\}$$

where

$$\{y\} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \qquad [X] = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix},$$

$$\{\beta\} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \text{and} \quad \{\varepsilon\} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_m \end{bmatrix}. \qquad \textbf{Equation 7}$$

where $\{y\}$ is an *m-by-1* vector of observations, $[X]$ is an *m-by-n* matrix of the independent variables, $\{\beta\}$ is an *n-by-1* vector of regression coefficients, and $\{\varepsilon\}$ is an *m-by-1* vector of random errors.

The least squares function is given by

$$\{L\} = \sum_{i=1}^{m} \varepsilon_i^2 = \{\varepsilon\}'\{\varepsilon\} = (\{y\} - [X]\{\beta\})'(\{y\} - [X]\{\beta\})$$

$$= \{y\}'\{y\} - \{\beta\}'[X]'\{y\} - \{y\}'[X]\{\beta\} + \{\beta\}'[X]'[X]\{\beta\}$$

$$= \{y\}'\{y\} - 2\{\beta\}'[X]'\{y\} + \{\beta\}'[X]'[X]\{\beta\} \qquad \textbf{Equation 8}$$

The components of the random error vector are minimized by

$$\left.\frac{\partial\{L\}}{\partial\{\beta\}}\right|_{\{b\}} = -2[X]'\{y\} + 2[X]'[X]\{\beta\} = 0$$

which , after simplification, becomes

$$[X]'[X]\{b\} = [X]'\{y\}. \qquad \textbf{Equation 9}$$

The vector $\{b\}$ represents the vector of least squares estimators of $\{\beta\}$ that minimizes *{L}*. Finally, $\{b\}$ is isolated by multiplying through by the inverse of $[X]'[X]$. This gives

$$\{b\} = ([X]'[X])^{-1}[X]'\{y\}. \qquad \textbf{Equation 10}$$

The fitted regression model is

$$\{\hat{y}\} = [X]\{b\}. \qquad \textbf{Equation 11}$$

In indicial notation, Equation 11 is

$$\hat{y}_i = b_0 + \sum_{i=1}^{n} b_j x_{ij}, \qquad i=1,2,\ldots,m \qquad \textbf{Equation 12}$$

### 4.1.2. Second order models

The techniques described above are easily adapted to approximate a second order model of the form

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^2 + \sum \sum_{i<j} \beta_{ij} x_i x_j + \varepsilon. \qquad \textbf{Equation 13}$$

In this case, the only difference lies in the formation of the [X] matrix. The minimum number of response observations needed to fit a second order model is given by

$$NS = \frac{(n^2 + 3n + 2)}{2} \qquad \textbf{Equation 14}$$

where $n$ is the number of design variables. Put another way, the length of $y$ must be at least $(n^2+3n+2)/2$. The first column of [X] corresponds to the $\beta_0$ term in all observations, so all entries in this column are set to unity. The columns 2 through $(n+1)$ correspond to the linear terms. Columns $(n+2)$ through $(2n+1)$ correspond to the pure quadratic terms while the interaction terms are located from columns $(2n+2)$ to $((n^2+3n+2)/2)$. Once the [X] matrix is formed, the vector $\{b\}$ is found in the same manner as above, and a quadratic response surface is easily attained.

The effect that dimensionality has on the total computational labor involved in generating response surfaces. Since the number of points needed to fit a quadratic increases with the number of design variables according to equation 14, efforts should be made to condense the number of design variables whenever possible. For a numerical example, consider a wing load distribution that is defined by forces at 10,000 discrete grid points on the wing surface in 2D coordinates (u,v). The approximate force p at a

point located at (u,v) is given by p = p(q,u,v). However, if p is formulated as a quadratic polynomial in q then the number of independent q is only 6 instead of 10,000. Then each q would be treated as if it were a separate coupling variable in Y*, Y^.

## 4.2. Error Analysis of Fitted Response Surfaces

For the current research, two tests were conducted to verify the accuracy of the fitted response surface. The first test involved the trivial case of fitting a quadratic response surface to a quadratic function. The second test involved the significantly more complex generalized Rosenbrock (Banana) function,

$$f(x) = \sum_{i=1}^{NX}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]. \qquad \textbf{Equation 15}$$

In both test cases, the response surfaces were generated using a random point placement scheme.

A simple error analysis was performed to quantify differences in the fitted response surface model and the actual functions. For both functions, response surface function evaluations were compared to actual function evaluations using the following:

$$\text{Normalized Error} = (F_{actual} - F_{RS})/F_{actual}. \qquad \textbf{Equation 16}$$

The results of this error analysis are reported in Figure 13 through Figure 16, where NX is the number of design variables, and NS is the number of points required to fit the quadratic. As expected, the error associated with approximating a quadratic function with another quadratic function is smaller than the error found in the approximation to the Banana function.

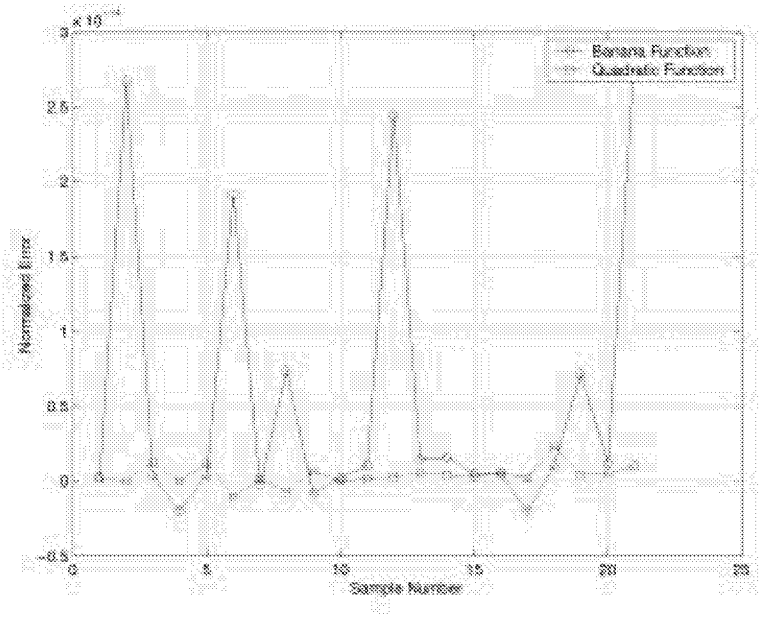


**Figure 13. Error analysis for NX=5, NS=21**

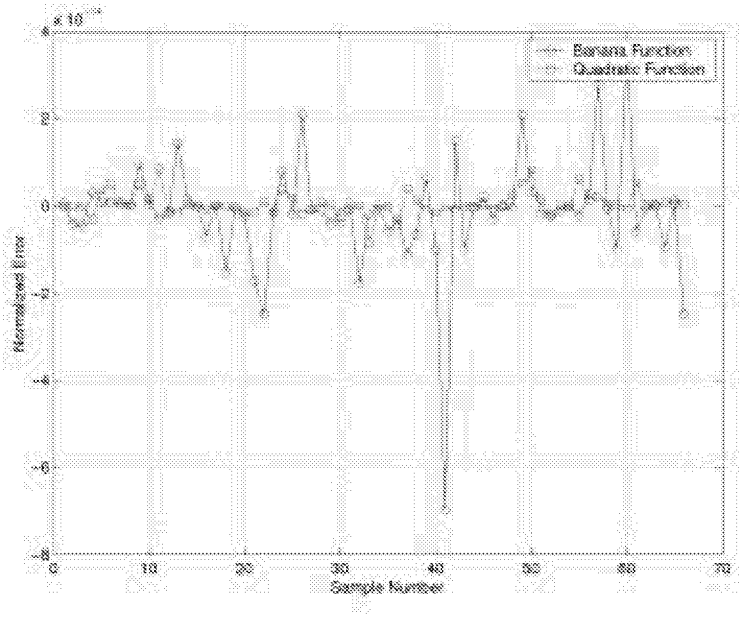


**Figure 14. Error analysis for NX=10, NS=66**

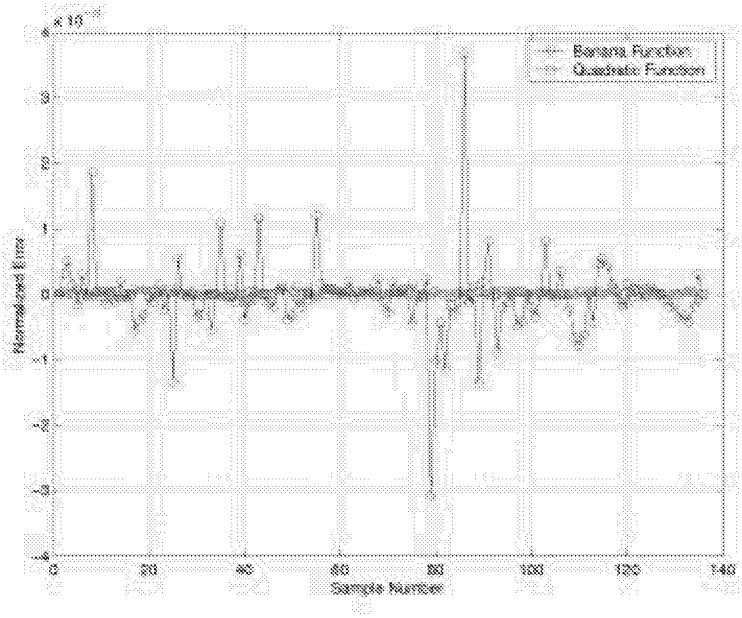


**Figure 15. Error analysis for NX=15, NS=136**

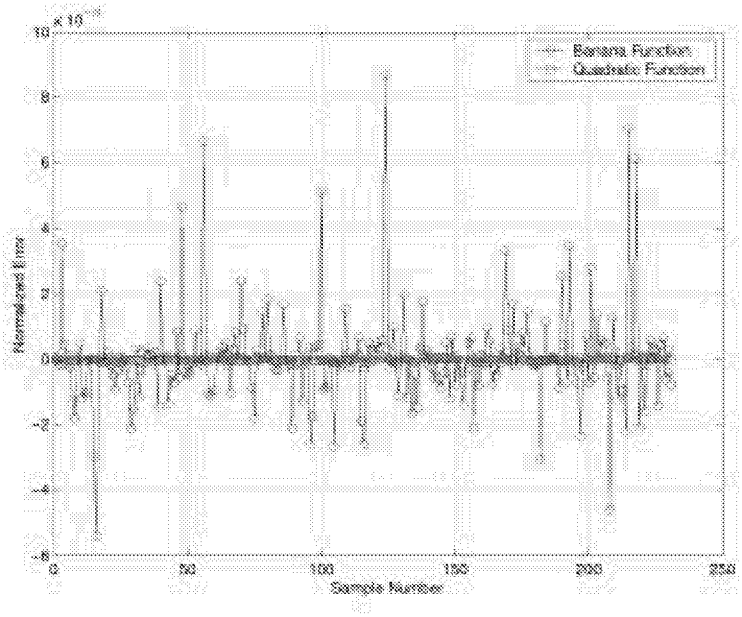


**Figure 16. Error analysis for NX=20, NS=231**

# 5. Design Point Placement

A number of design point placement techniques were examined to determine the most efficient way to explore the design space. Since the BB analyses were somewhat computationally costly (especially in the case of structures and aerodynamics), and since there were a high number of input variables to some BB's, traditional DOE patterns such as central composite or Box-Behnken were impractical. This led to the consideration and testing of some alternative point placement schemes: Hypersphere, Monte Carlo (random) and D-optimal Point Placement schemes. In accordance with DOE literature, the term *design* used in a DOE context refers to a particular point dispersion technique, and should be differentiated from design in the physical sense.

## 5.1. Hypersphere Point Placement

The first DOE scheme investigated involved points placed uniformly on a hypersphere in n-dimensions, coupled with an anti-bunching mechanism. The hypersphere placement of points reduces the likelihood of point bunching, a common phenomenon (especially in the corner locations) of a hypercube design. Additionally, since the nature of the experiments performed was deterministic, the number of points placed on the hypersphere surface was significantly lower than that of a CCD or Box-Behnken design. In fact, the number of points placed using the hypersphere is governed only by the minimum number of points necessary for a second order surface fit to the data.

The method used for generating uniformly distributed random points on the surface of the hypersphere was adapted from [15]. To protect against the possibility of a

point falling very closely to or on top of its nearest neighbor, an "anti-bunching" mechanism was developed. Overlapping points would in effect act like a single point, therefore reducing the total number of points by one. Since the goal here is to generate the fewest points required to fit a quadratic surface, an overlapping set of points would drop the number of points below the minimum required for a quadratic response surface fit. Additionally, the accuracy of the fitted surface is reduced when the points are bunched together. A complete description of the hypersphere point placement with antibunching algorithm can be found in Appendix A.

Once implemented, the hypersphere point placement method appeared to exhibit unacceptable biasing toward the center of the design space, especially for models of higher dimension [16]. This can be explained by examining Figure 17 through Figure 19.

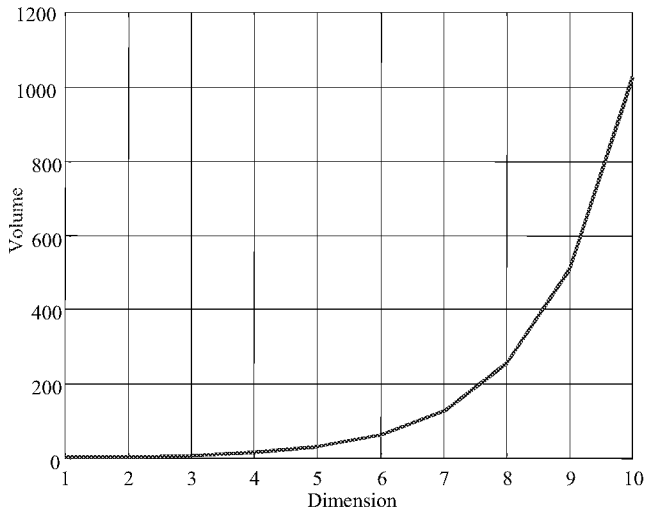


**Figure 17. Volume of N-Dimensional Hypercube [16]**

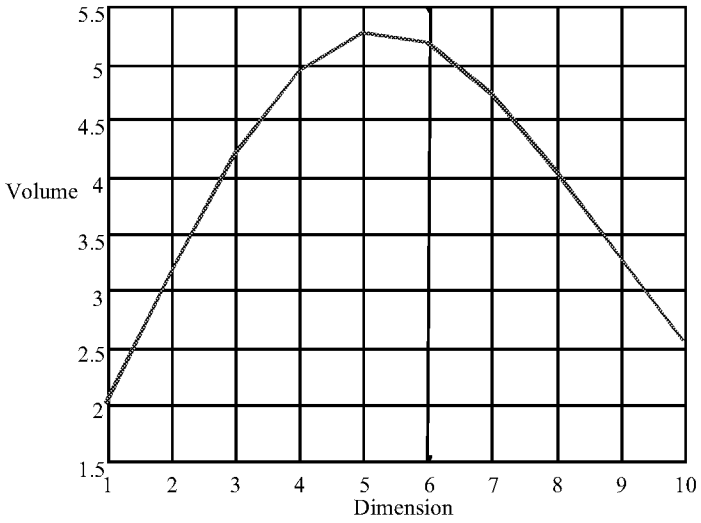


**Figure 18. Volume of N-Dimensional Hypersphere [16]**

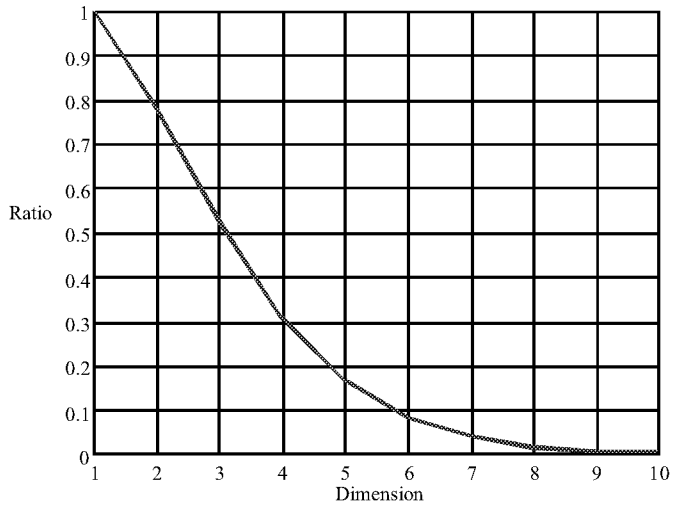


**Figure 19. Ratio of Hypershpere Volume to Hypercube Volume [16]**

Contrary to a cube in N-dimensions, a sphere does not continually increase in volume with increasing dimensionality. In fact, after as little as N = 5 the volume of the N-dimensional hypersphere actually decreases, resulting in the ratio of hypersphere volume to hypercube volume quickly approaches zero. One can safely deduce that the volume of the hypersphere at high dimensionality does not effectively overlap a

comparable hypercube volume. Thus, the design space of interest is poorly approximated by the hypersphere point placement scheme. The hypersphere point placement scheme was ultimately abandoned for these shortcomings. A detailed formulation for the volume of a hypersphere can be found in Appendix A.

## 5.2. Random Point Placement

The next point placement scheme involved a random Monte Carlo generation of input variables to each BB. The Matlab random number generator was used to generate a random vector $\xi$, whose elements lie between 0 and 1. Once mapped into [-1 1] coordinates, the random vectors of inputs were used as parameters for local optimization and formation of the response surface.

The random point placement was extremely simple to implement and showed good dispersion of the input parameters. However, once the response surfaces were fit and examined, there was evidence that these quadratic models gave a poor representation of the actual black box, especially at the edges of the design space. The origin of this disparity could come from two possible sources. First, the minimum number of points required to fit a quadratic may have simply yielded a poor fit. Secondly, the extremities of the response surfaces appeared not to "anchor" themselves at the end bounds of the data. Of course, it is impossible to plot an $n^{th}$ dimensional surface, so these hypotheses are based on slicing the design space and viewing contour plots of various input combinations.

For the reasons stated above, efforts were made to investigate the effects of using more than the required minimum number of points for a quadratic fit. Indeed, by increasing the number of points, the correlation of the RS surrogate to the actual BB

response was improved. However, it was found through experience that reducing the intervals over which RS's were generated yielded better results.

## 5.3. D-Optimal Point Placement

To address the shortcomings of the random point placement scheme, a D-Optimal method was considered. The D-optimal design belongs to a class of computer generated designs first formulated in the 1970s. A D-optimal design is one that maximizes the determinant of $X^{T}X$. It can be shown that $det(X^{T}X)$ is inversely proportional to the square of the volume of the confidence ellipsoid of the regression estimates of the linear model [12]. The volume of this confidence region is important because it reflects how well the set of coefficients are estimated.

The `cordexch` (coordinate exchange) function in Matlab's statistics toolbox was chosen to generate the D-optimal design. This is an iterative algorithm that operates by improving a starting design by making incremental changes to its elements [17]. The increments are the individual elements of the design matrix. `cordexch` requires the user to specify the number of inputs, the number of runs or design points, and the order of the model. An important feature of this type of design is that the user can specify the number of points placed, so long as that number is large enough to fit a second-order response surface through the data.

Baseline response surfaces are critical in subsequent BLISS iterations because the design space is reduced, then focused on the region that contains the system optimum. Therefore, if a reduced interval does not contain the system optimum, poor results will encompass all other iterations. Ultimately, the baseline design that was chosen was a D-optimal design. This choice seemed most appropriate since it efficiently searched the

design space with a limited number of design points. The D-optimal design also seemed to provide reasonable consideration to the edges of the design space – an attribute common to CCD or Box-Behnken designs, but made possible with far fewer design points.

Since BLISS-RS calls for the response surfaces to be fit through locally optimum data, it was imperative that all local optimizations were successfully converged. Therefore, a provision was implemented that allowed for the placement of a random design point whenever a particular D-Optimal design point failed to converge successfully. Such random points were generated until one was found that successfully converged. Thus, the D-Optimal design was ultimately augmented by the placement of randomly dispersed points

## 5.4. Coded and Natural Variables

A coordinate mapping is used to relate the coded variables generated by the point placement algorithm (ranging from –1 to 1) to the natural values of the variables in the physical problem. Figure 20 shows how such a mapping can be used to specify the natural coordinates in terms of their coded counterparts.

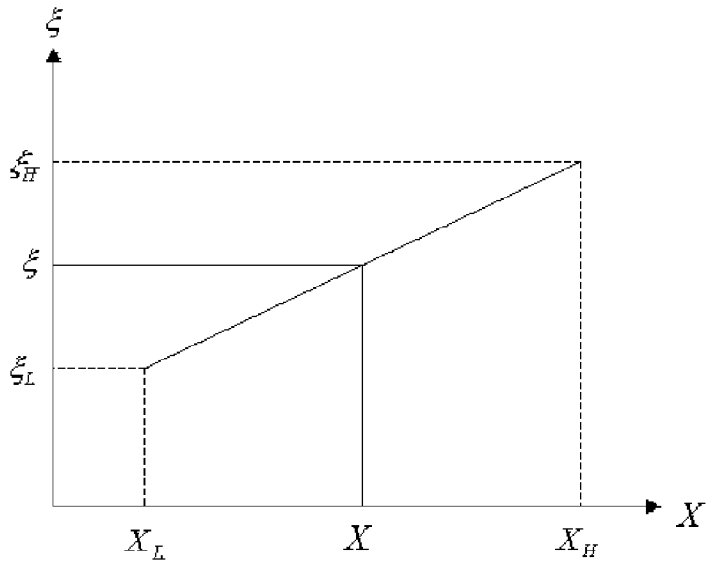


**Figure 20. Linear mapping for Coded and Natural Variables**

In equation 17, $\xi$ is the coded coordinate of a design point while $X$ is the actual value. Notice that the high and low values for natural variables must be specified beforehand. These values may depend on the physical constraints of the problem or may simply be the region of interest for the experiment (e.g. the exploration interval).

$$\frac{\xi - \xi_L}{\xi_H - \xi_L} = \frac{X - X_L}{X_H - X_L} \qquad \textbf{Equation 17}$$

# 6. Interval Reduction

Once a system optimization is conducted, it is important to check the accuracy of the response surface fit. A simple calculation was used to compare values taken from the response surfaces to those produced directly from the BB analyses. First, the values {Z | Y | w}* are found by the system optimization. Then these values were used as BB input parameters, and the BB's were re-optimized. The difference $Y_{BB}$*-$Y_{RS}$* was taken and

normalized by $Y_{BB}$* to give a measure of the "goodness of fit" obtained in the quadratic response surface model.

The measure of error due to lack of fit was used as a termination criteria. If, at the end of one iteration, the values $Y_{RS}$* and $Y_{BB}$* do not agree by a predetermined tolerance, the response surfaces are reformulated by reducing the intervals around system optimal values.

Reducing the interval sizes has the general property of improving the fit of the response surface. Indeed, as the intervals are reduced, the fit becomes more and more linear. It is desired that the new interval be centered on the anticipated optimum value, and that the new interval makes physical sense. Since the Y and w variables are not subject to any clear physical limitations, their interval bounds are allowed to "drift" across the original interval bounds. However, the Z variables are subject to such physical limitations so their intervals must always lie within the original interval bounds. These differences lead to the use of two different interval reduction methods.

Intervals on system level variables Z are reduced in such a way as to ensure the subsequent interval would fall within the physical bounds described by the original interval. For example, let $XL_i$ and $XH_i$ denote the respective lower and upper limits on the interval used to generate a response surface. If the termination tolerance is not satisfied, then the interval is reduced by 20% and centered on the system optimal value. If any of the new $XL_{i+1}$is less than $XL_i$ , the value is returned to $XL_i$. Similarly, if any of the $XH_{i+1}$ is larger than $XH_i$ the interval limit is returned to $XH_i$. This operation shrinks the interval over which points are placed to generate the response surfaces. It should be

noted, however, that reducing the intervals in this manner does not ensure that the new interval is centered on the optimum found in the previous iteration.

The factor by which the interval is reduced each iteration was chosen somewhat arbitrarily. Since the interval is reduced each iteration (without the possibility of being increased), it is important to reduce the interval size gradually to prevent premature convergence to an unreliable optimum

The intervals for coupling variables Y and weight factors w were not subject to the limitation of falling within the original interval. Therefore, the intervals are allowed to shift solely on the basis of where the optimum was found on the previous iteration. This carries the added benefit of centering the RS's in the middle of the design space with respect to Y and w, so satisfaction of the coupling constraints is always possible.

A simple algorithm was adapted from [16] for re-sizing the intervals on Z and w. First, a factor K is defined for reducing the interval size. This factor is not directly applied to the current interval. Instead, an adjustment factor $A_i$ is determined by linear interpolation as:

$$A_i = \frac{|C_i - Opt_i|}{C_i - L_i} \qquad \textbf{Equation 18}$$

where $C_i$ is the center of the current interval, $Opt_i$ is the optimum found in the current interval, and $L_i$ is the lower value of the current interval. The actual reduction factor can then be found as:

$$R_i = A_i + (1 - A_i)K \qquad \textbf{Equation 19}$$

The result of this process is that the interval size will not be reduced if the optimum point is located on a boundary, while it will be reduced by a factor of $K$ when the optimum point is located at the center.

Additionally, a history of optimum location with respect to bounds is kept. If the best design point hits the same bound twice, the interval is expanded by a fixed amount. If the optimum oscillates from upper to lower bounds between iterations, the interval is reduced by a fixed amount.

# 7. BLISS-RS Results

BLISS-RS was tested on the design of a supersonic business jet from a 1995 AIAA Graduate Student Design Competition. The baseline model is shown in Figure 21 with corresponding dimensions listed in Table 1.

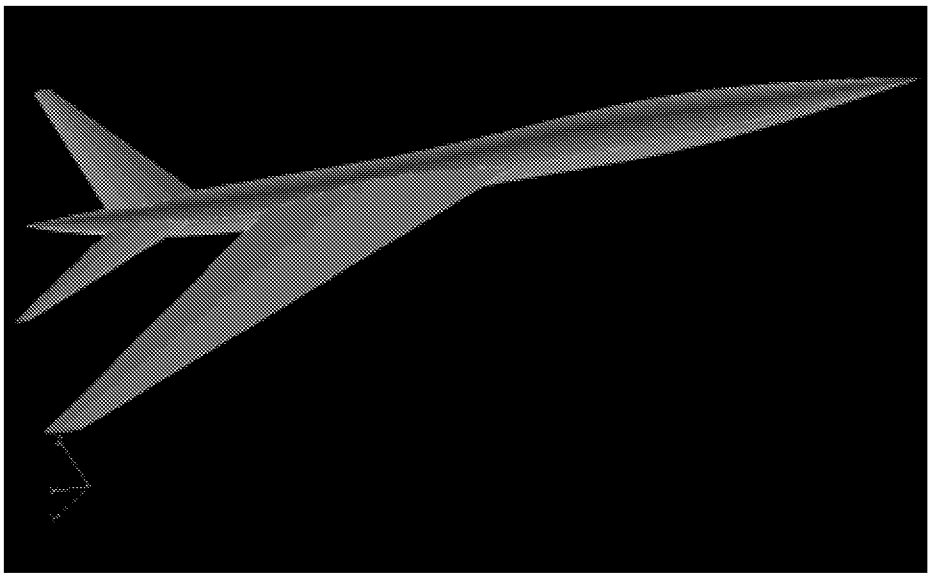

**Figure 21. Baseline Model of Supersonic Business Jet**

**Table 1. Baseline Geometrical Variables**

| Variable Name | Symbol | Initial Low Value | Initial Value | Initial High Value | Units |
|---|---|---|---|---|---|
| thickness ratio | t/c | .01 | 0.075 | .1 | -- |
| wing aspect ratio | $AR_W$ | 3.5 | 4 | 6.5 | -- |
| wing sweep | $\Lambda_W$ | 40 | 45 | 70 | degrees |
| wing surface area | $S_W$ | 200 | 400 | 800 | $ft^2$ |
| tail surface area | $S_{HT}$ | 40 | 120 | 125 | $ft^2$ |
| tail aspect ratio | $AR_{HT}$ | 3.5 | 4.5 | 5.5 | -- |
| taper ratio | $\lambda$ | .1 | 0.2 | .4 | -- |

For each BB, equation 14 was used to find the minimum number of points required for a quadratic RS. Table 2 lists the variables Z, Y*, and w that were input to each BB, and the corresponding number of design points placed. The weighting factors are labeled in such a way as to denote the BB number immediately following w, and the output number in parentheses. The weighting factor of one of the outputs of each BB was held constant (except for the performance BB since there is only one output). This imparts relative importance to the other weighting factors while also reducing the total number of variables. These were chosen as w1(1), the weighting factor for $W_T$, w2(2), the weighting factor for drag, and w3(2), the weighting factor for $W_E$, since it was obvious that minimizing these terms would have a beneficial effect on range.

**Table 2. RS Generation Data**

| BB | Output | Input Variables | | | Number of Inputs | Number of Points (NS) |
|---|---|---|---|---|---|---|
| | | Z | Y* | w | | |
| 1. Structures | 1. $W_T$<br>2. $W_F$<br>3. $\Theta$ | t/c, $AR_W$, $\Lambda_W$, $S_{REF}$, $S_{HT}$, $AR_{HT}$, $\lambda$ | L, $W_E$ | w1(2)<br>w1(3) | 11 | 78 |
| 2. Aero | 4. L<br>5. D<br>6. L/D | t/c, h, M, $AR_W$, $\Lambda_W$, $S_{REF}$, $S_{HT}$, $AR_{HT}$, $\lambda$ | $W_T$, $\Theta$, ESF | w2(1)<br>w2(3) | 14 | 120 |
| 3. Power | 1. SCF<br>2. $W_E$<br>3. ESF | h, M | D | w3(1)<br>w3(3) | 5 | 21 |
| 4. Performance | 1. Range | h, M | $W_T$, $W_F$, L/D, SFC | | 6 | 28 |

For simplicity, the procedure was terminated after ten iterations. It was decided from previous executions that this number of iterations sufficiently demonstrated the convergence characteristics of the system without excessive run time.

The total labor for each BB in Table 2 is NS times the number of iterations for convergence, the latter being 10 for the current research. If a concurrent processing environment were used, the elapsed time for each BB would be equal to the time for one NS generation. Therefore, the total elapsed time to generate all RS data would be equal to the elapsed time for the longest single BB optimization, and would scale linearly in proportion to the above longest time.

Figure 22 through Figure 30 show the location of the optimum value for each of the system-level variables over the 10 iterations. Also shown are the intervals over which the response surfaces were formed per iteration. It is clear that by forcing all subsequent intervals to remain inside the original interval, some of the $(i+1)^{th}$ intervals are not centered on the $i^{th}$ optima.

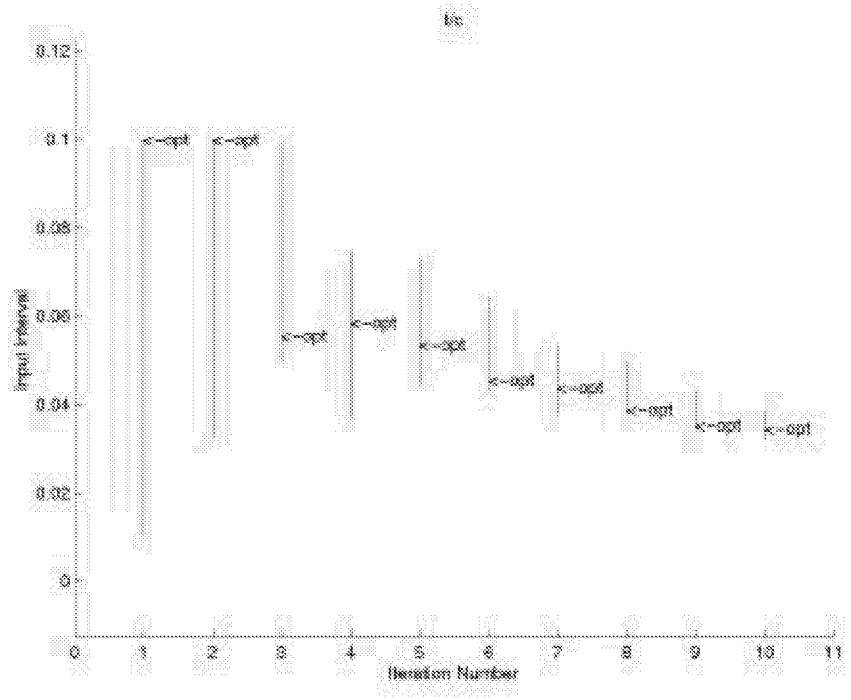


**Figure 22. Optimization History for t/c**

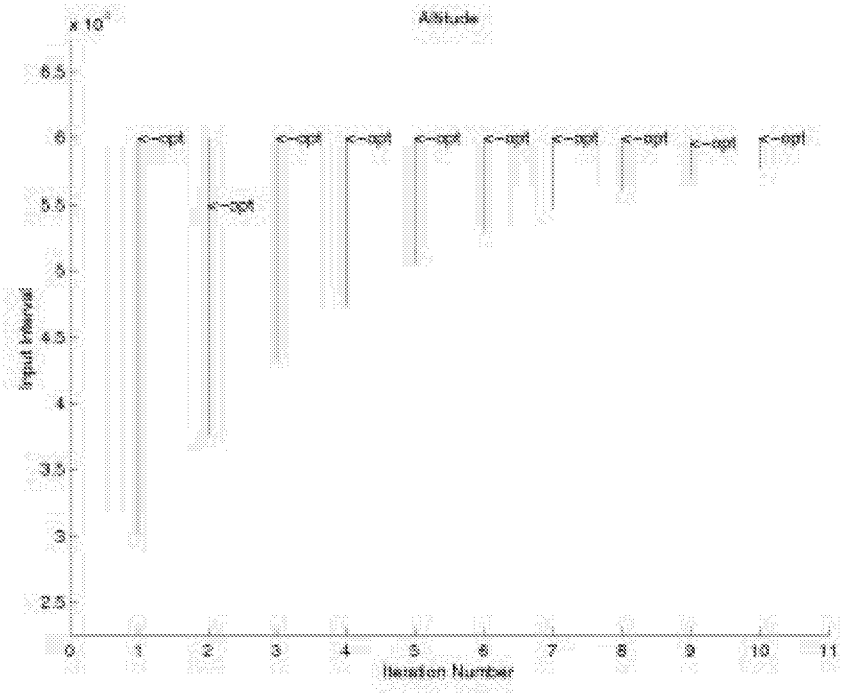


**Figure 23. Optimization History for Altitude**

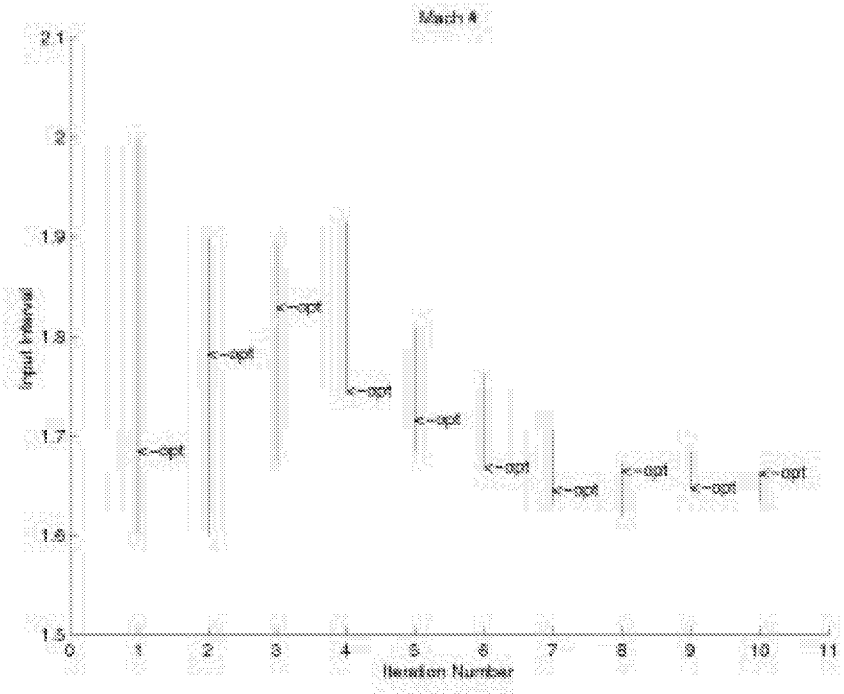


**Figure 24. Optimization History for Mach Number**

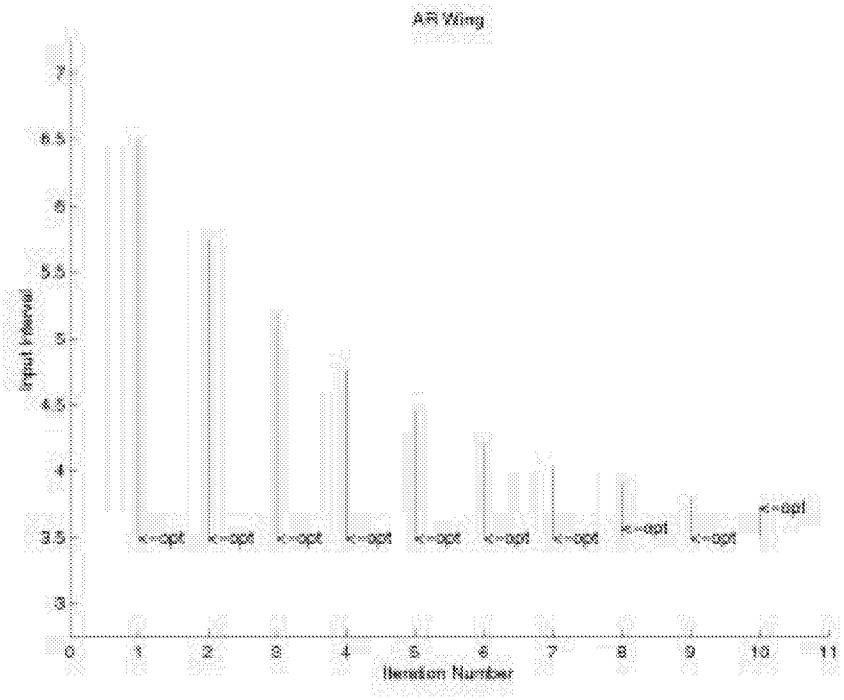


**Figure 25. Optimization History for Wing Aspect Ratio**

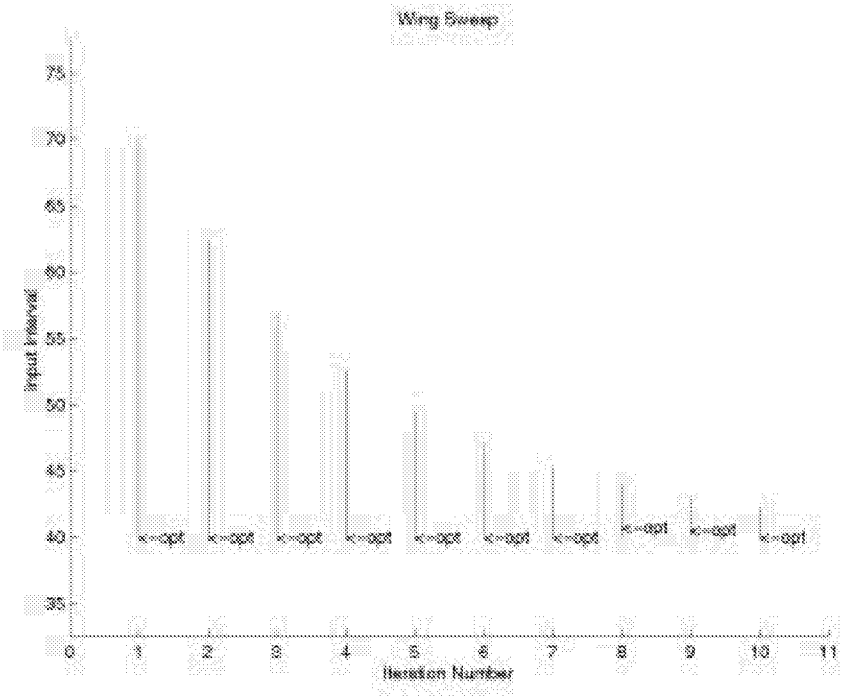


**Figure 26. Optimization History for Wing Sweep (deg)**

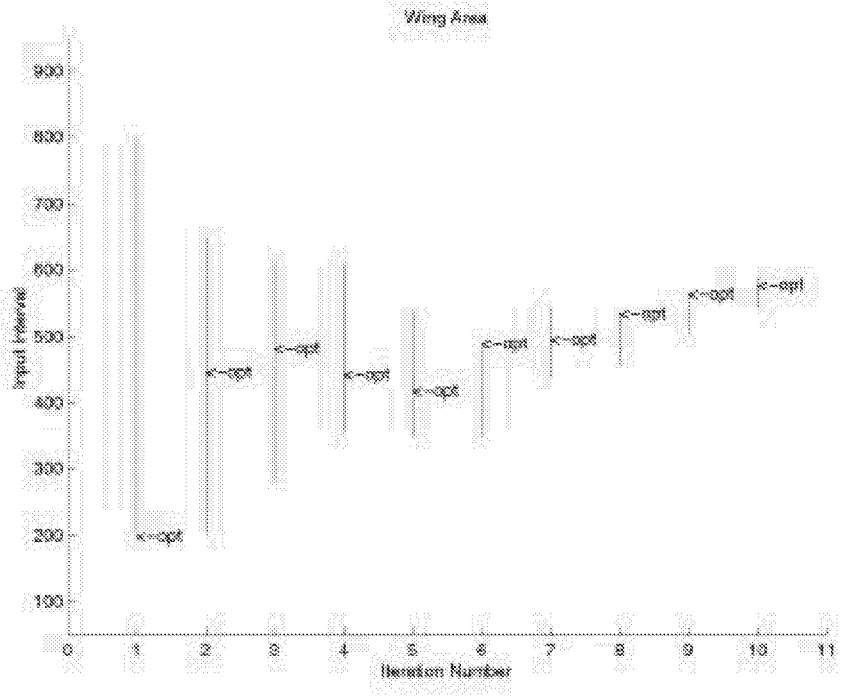


**Figure 27. Optimization History for Wing Area ($ft^2$)**

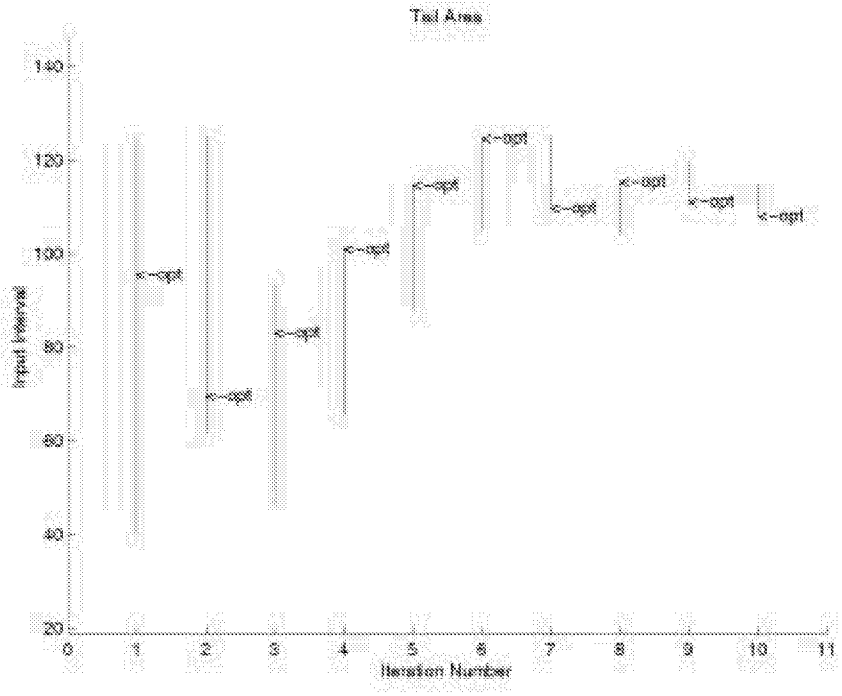


**Figure 28. Optimization History for Tail Surface Area**

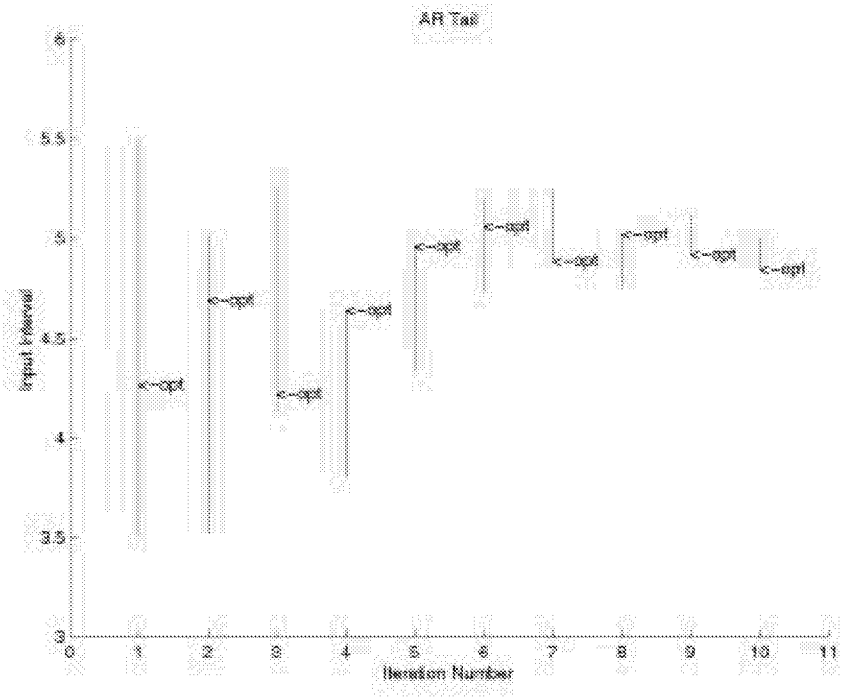


**Figure 29. Optimization History for Tail Aspect Ratio**

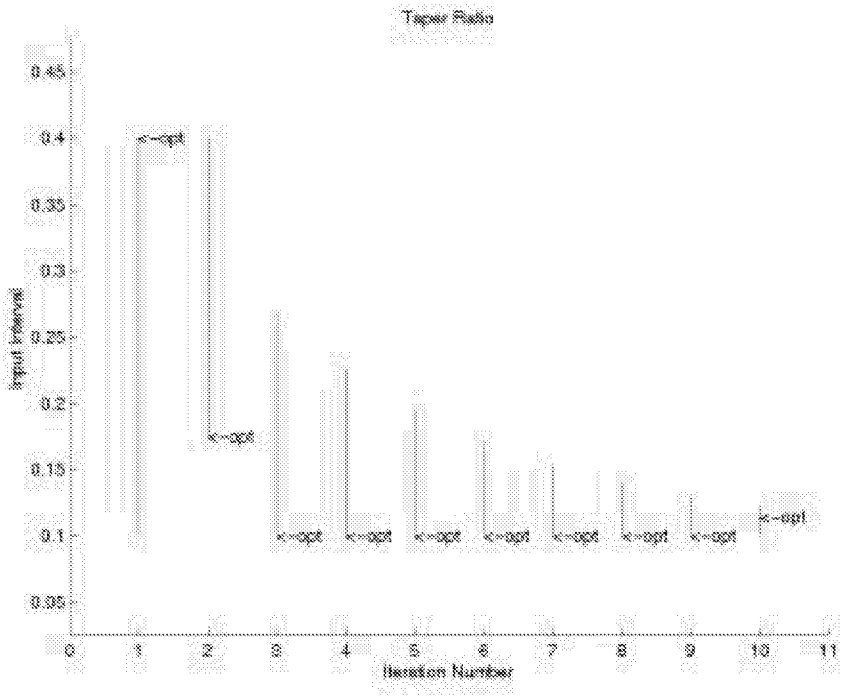


**Figure 30. Optimization History of Taper Ratio**

Similarly, Figure 31 through Figure 39 show the location of the optimum value for each of the coupling variables, along with interval size for 10 iterations. In this case, the intervals are allowed to "drift" across the original interval bounds, and allows the $(i+1)^{th}$ interval to be centered on the $i^{th}$ optimum in all cases. This is a helpful characteristic when it comes to satisfying the coupling constraints {c}=0. The figures also demonstrate the workings of the algorithm given in equations 18 and 19. Intervals are reduced more aggressively if the optimum falls at the center of the interval (a factor of 3 was used here). The intervals were expanded by a factor of two if the optimum landed at a boundary for two consecutive iterations, and the interval was reduced by a factor of two if oscillation occurred.

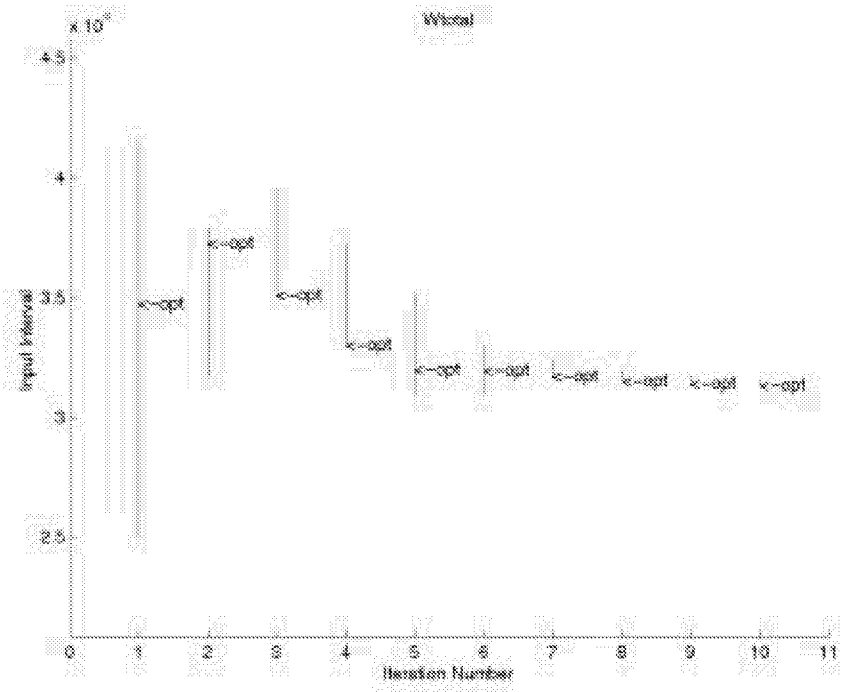


**Figure 31. Optimization History for Total Aircraft Weight**

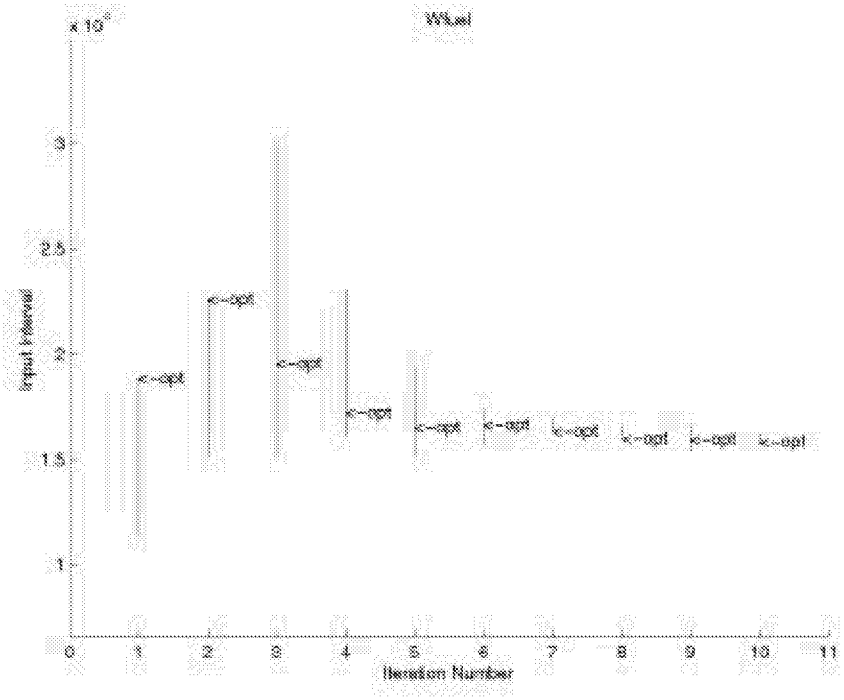


**Figure 32. Optimization History for Fuel Weight**

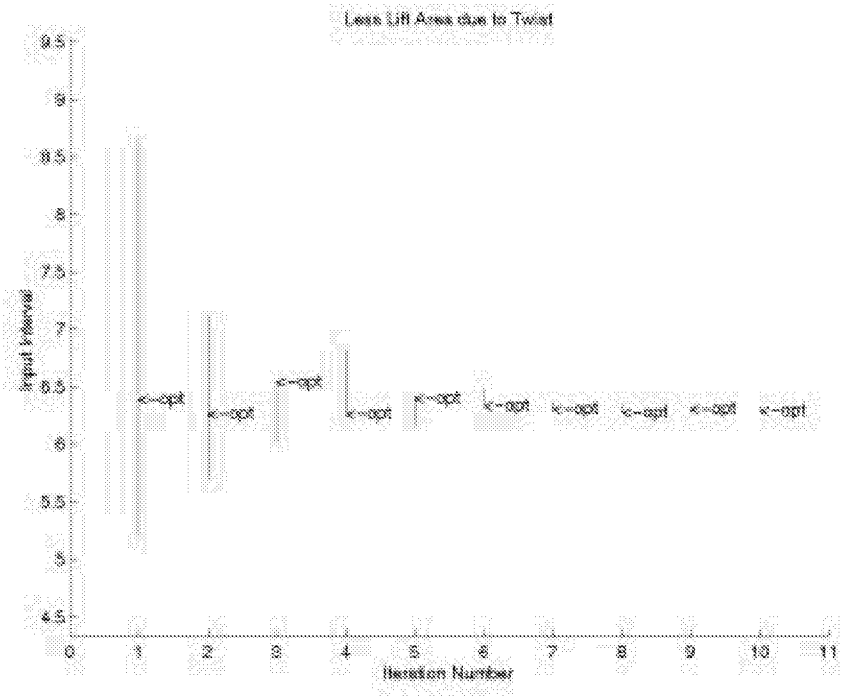


**Figure 33. Optimization History of "Twist"**

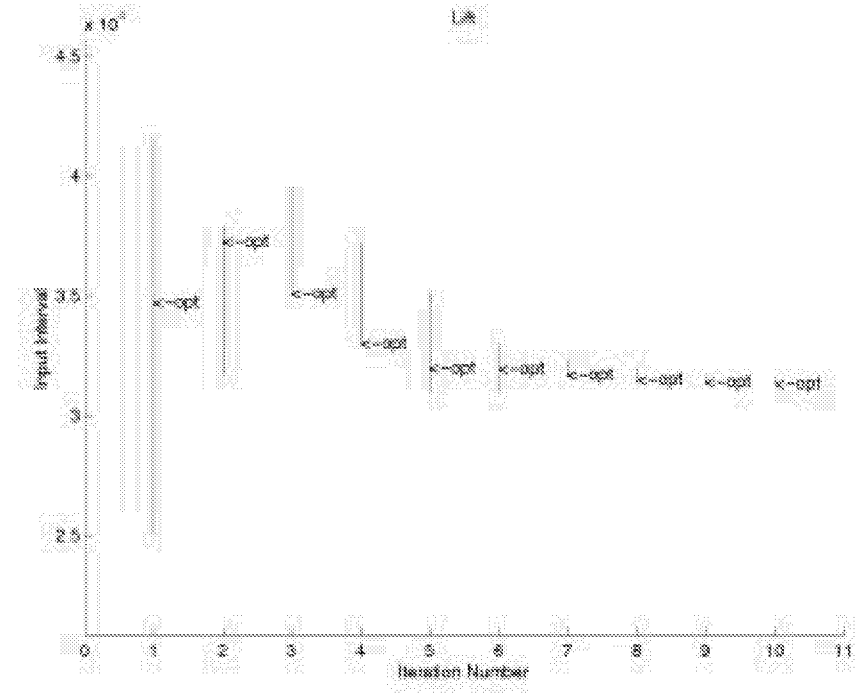


**Figure 34. Optimization History for Lift**

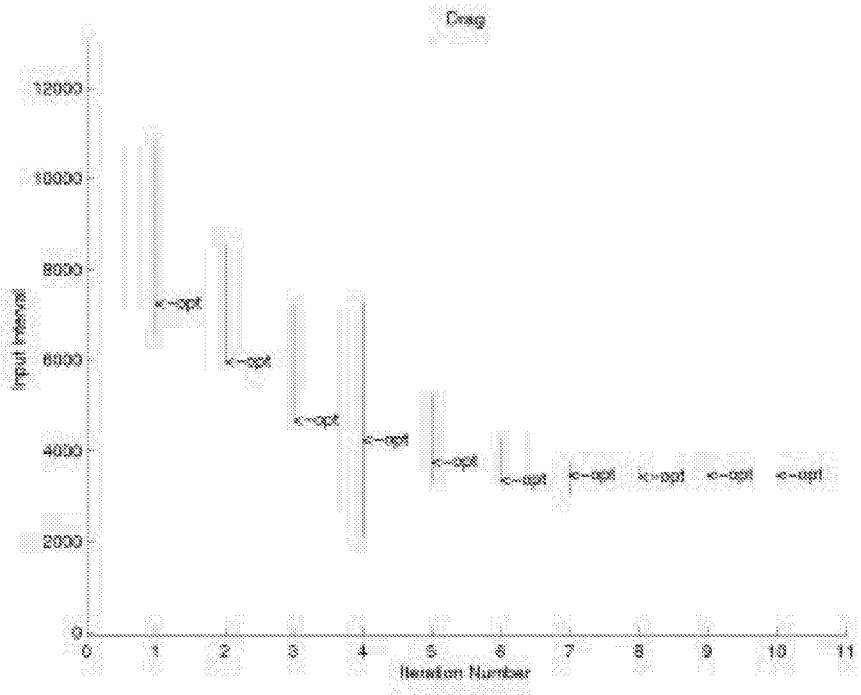


**Figure 35. Optimization History for Drag**

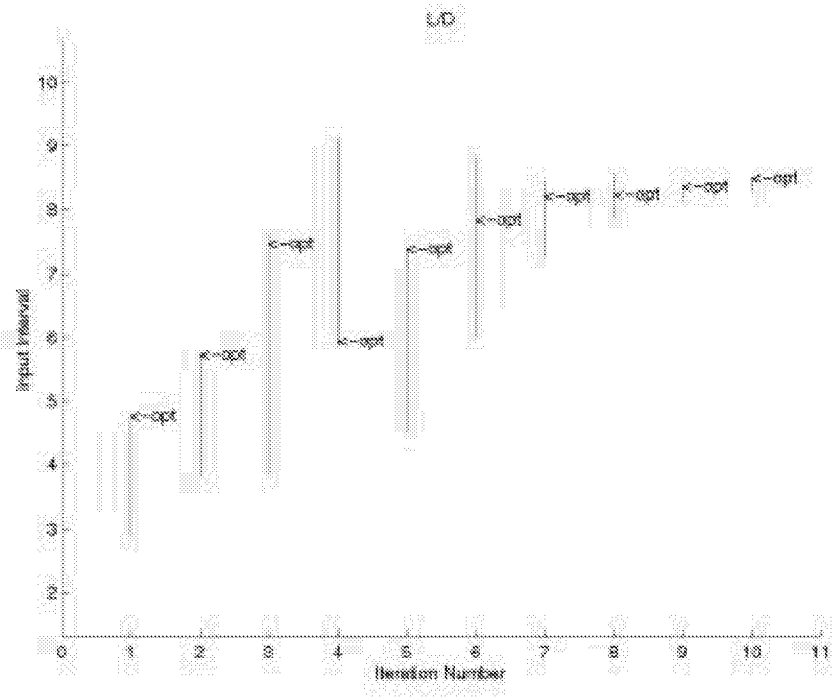


**Figure 36. Optimization History for L/D**

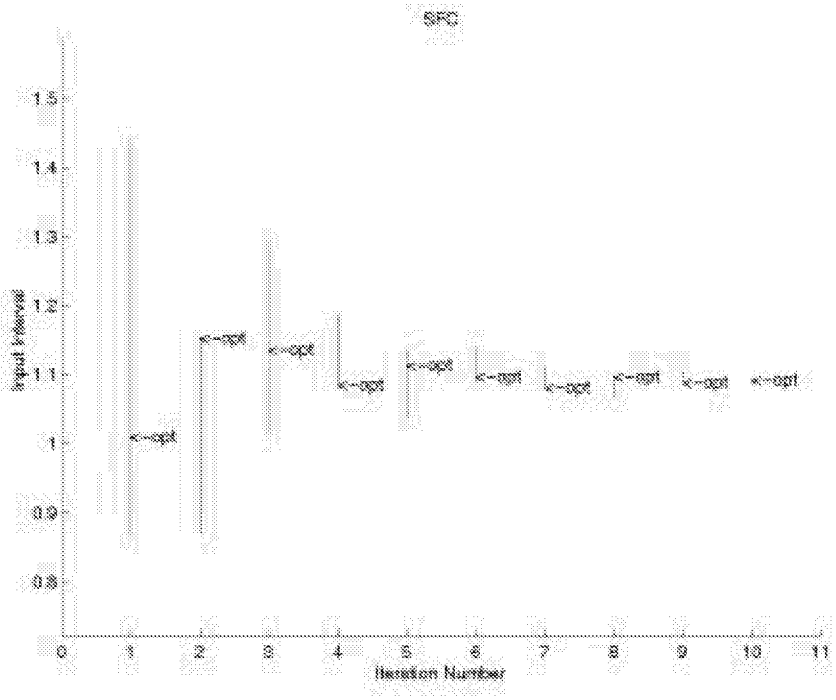


**Figure 37. Optimization History for SFC**

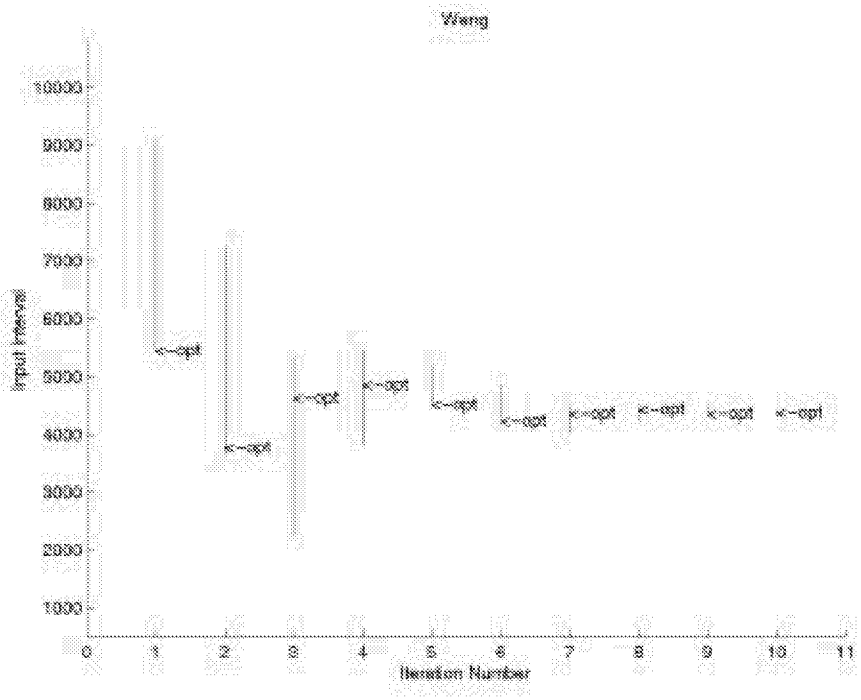


**Figure 38. Optimization History for Engine Weight**

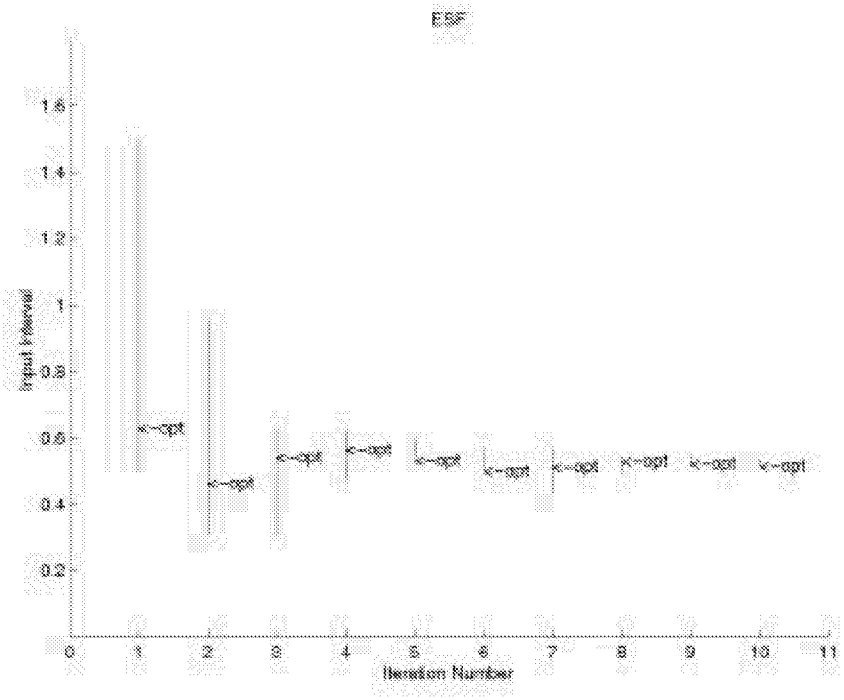


**Figure 39. Optimization History for ESF**

The weighting factors are all plotted in Figure 40 for direct comparison. The values that the weighting factors converge to reflect the influence they have on the system objective. For instance, the weighting factor for lift, w2(1), is about –1, indicating that lift is being maximized (for the standard that positive numbers are minimized). Likewise, the weighting factor for SFC, w3(1), is 0.5, meaning SFC is being minimized. Both results are intuitively correct.

Figure 40 also shows that the weighting factors for fuel weight, w1(2), and for ESF, w3(3), have not completely converged. This is clearly a product of the termination criteria chosen.

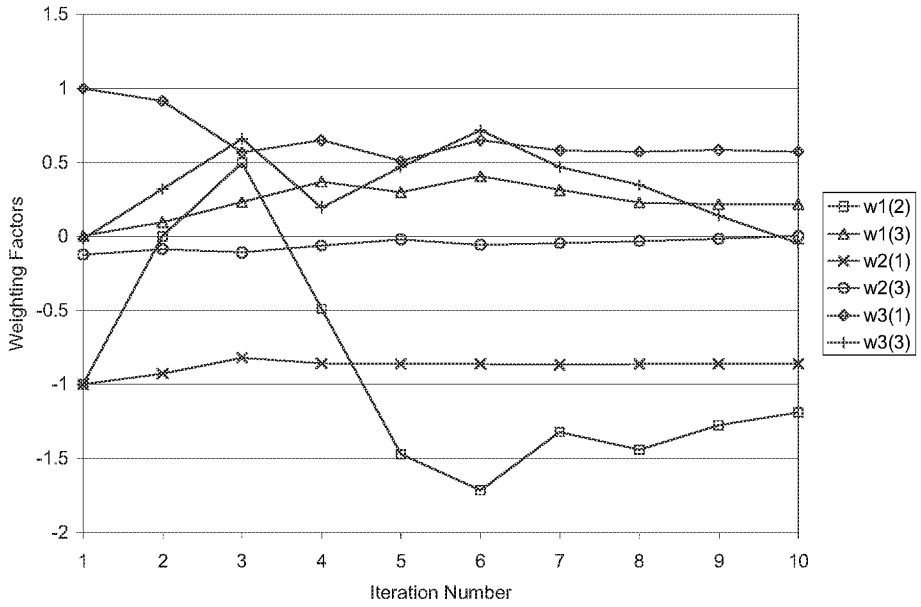


**Figure 40. Optimization History for Weighting Factors, w**

The optimized ranges found by the RS representations and BB's for each iteration are shown in Figure 41. In general, the response surface values closely resemble those obtained from the actual black box. The range of an actual aircraft would be substantially lower than those in shown in Figure 41. This is expected since constraints such as takeoff run length, climb rate requirements, engine-out conditions, and several aeroelastic limitations were not considered.

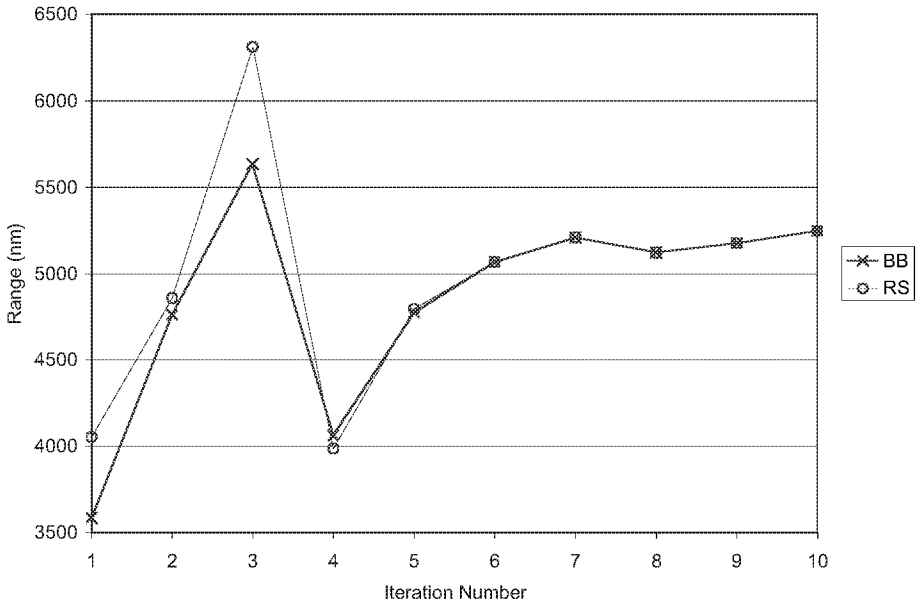


**Figure 41. Range History**

In order to assess the validity of the response surface models, the system level variables obtained in system optimization were used to re-optimize each BB. The corresponding outputs $Y_{BB}$ from this operation were compared to the $Y_{RS}$ obtained from the system optimizer. This comparison can be seen in Figure 42.

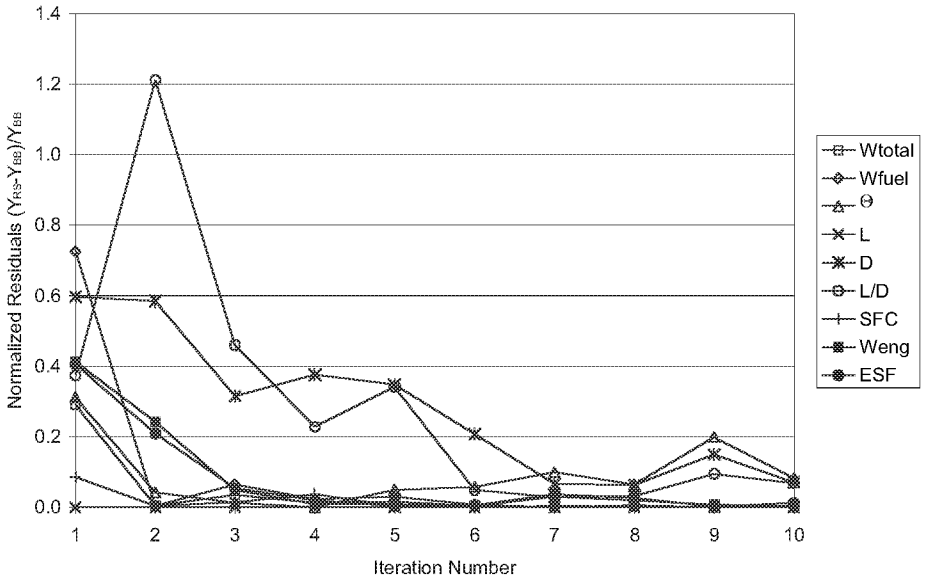


**Figure 42. Plot of Residuals for Coupling Constraints**

The data in Figure 42 were all normalized by the BB value in order to make a more direct comparison for the various coupling variables. As this graph shows, the interval shrinking technique does (in general) improve the accuracy of the response surface model. However, use of this method does not insure that each subsequent iteration will yield a better fit than the one before it.

It is important to point out here that the comparison of the RS values to the BB values in Figure 42 is somewhat limited in scope. Ideally, the local variables should be represented by response surfaces of their own in order to show their behavior over the specified input intervals. One could then pull the information regarding local variables at the system optimum from these RS's instead of re-optimizing locally. Unfortunately, this would require a very large amount of data to be carried along through the BLISS process, and was therefore not performed.

Figure 43 through Figure 45 show the optimization histories for some local variables for structures, aerodynamics, and power black boxes, respectively. In Figure 43, the skin caliper thickness $t_{s1}$ through $t_{s3}$ are defined via Figure 6. The thickness are labeled "inner" to denote the spanwise location of the panels (inboard, or closest to the fuselage in this case). This figure also indicates, the values for $t_{s1}$ (inner) and $t_{s3}$ (inner) overlap. Additionally, the values for all three local variables shown do not deviate much from the initial values. Both anomalies can be attributed to the relatively relaxed termination criteria chosen for the structures BB. This was done to speed up the construction of response surfaces.

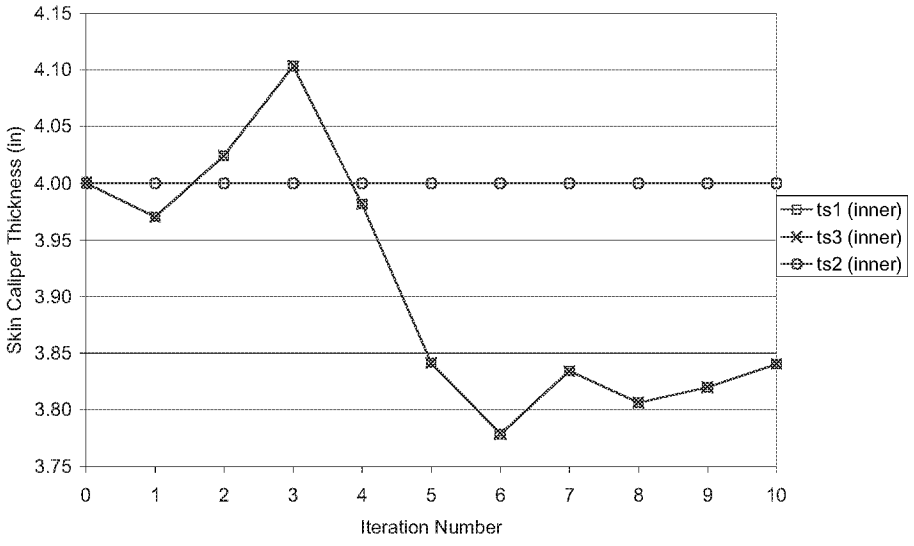


**Figure 43. Iteration Histories for Typical Structures BB Local Variables**

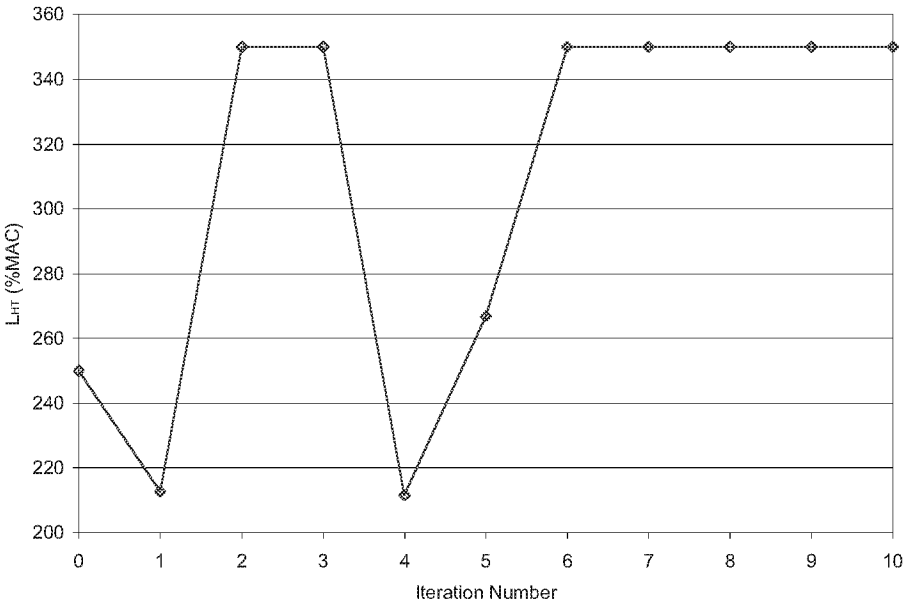


**Figure 44. Iteration History for Aerodynamics BB Local Variable $L_{HT}$**

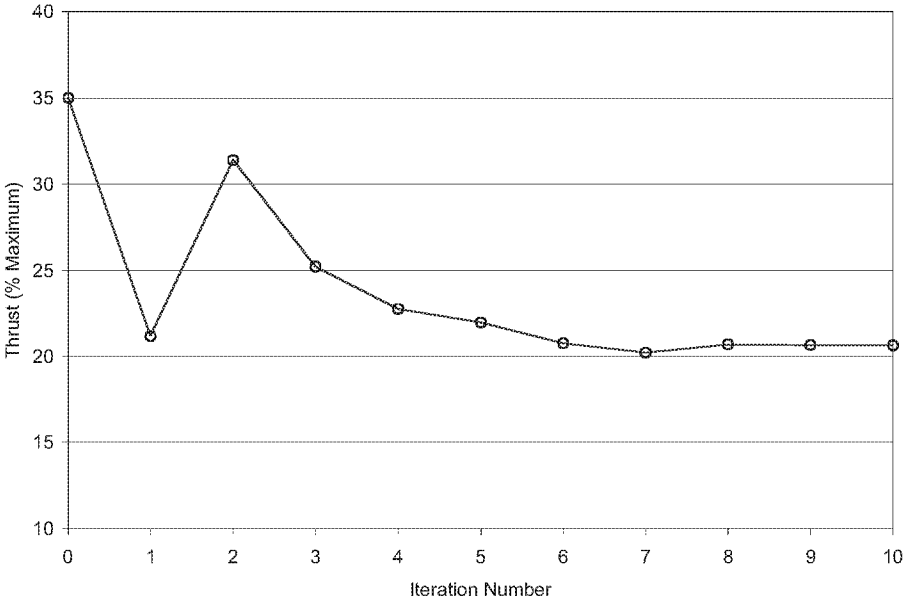


**Figure 45. Iteration History for Power BB Local Variable Thrust**

# 8. Black Box Fidelity Improvements

Since BLISS is modular by nature, the analyses included within each of the black boxes may be readily replaced. Indeed, the number of black boxes is also problem dependent. It is in this capacity that considerable efforts have been undertaken to demonstrate such capabilities. The following sections describe auxiliary work gone into BLISS-RS to improve the fidelity of submodule analyses. However, it should be noted that the results presented earlier do not pertain to these improvements.

The most prominent improvement is the inclusion of a computational fluid dynamics package for aerodynamic analysis. This improved aerodynamics BB facilitates changes within the inner workings of the structures BB. The power and performance black boxes remained exactly the same as in the previous version of BLISS, aside from some trivial input/output modifications.

## Improved Aerodynamics Black Box

The fidelity of the aerodynamics black box was improved by employing a computational fluid dynamics (CFD) code, coupled with grid-morphing software. Since the RS methodology calls for an optimization of local variables at each of the proposed design points and optimization of any local variables would be extremely expensive computationally, it was decided that all local variables in the aerodynamics BB be eliminated. Thus the new aerodynamics BB is purely an analytical tool for finding the desired aerodynamic loads, namely Lift and Drag.

The CFD code chosen for this task was CFL3D version 6.0, short for Computational Fluids Laboratory 3-Dimensional flow solver. CFL3D was originally

developed at by the Computational Fluids Laboratory at NASA Langley Research Center in the early 1980's [19].

CFL3D solves the time-dependent conservation law form of the Reynolds-averaged Navier-Stokes equations. Some of the features of CFL3D are outlined below:

- Semi-discrete finite volume approach for special discretization
- Up-wind biasing for convective and pressure terms
- Central differencing for shear stress and heat transfer terms
- Implicit time advancement with ability to solve for steady or unsteady flows
- Multigrid and mesh sequencing available for convergence acceleration
- Multiple turbulence models available for 0,1,or 2-equation models
- Multiple-block toloplogies possible through 1-1 blocking, patching, overlapping, and embedding.

CFL3D does not include any grid-generation software, so it was necessary to obtain the appropriate grid prior to execution. The baseline grid was generated at GEOLAB (Geometry Laboratory) at NASA Langely Research Center. The lab provides production and consultation services for computer aided design and numerical grid generation for various research fields. The baseline model chosen for BLISS-RS was that of the supersonic business jet used in previous BLISS work. The particular geometric parameters were obtained from the results of BLISS-98 optimization.

Since each point on a RS represents a unique design, a unique surface and volume grid must be used for analysis at that point. Therefore, instead of reproducing the grid each time a new design was proposed, a grid-morphing program was used. The MASSOUD algorithm [20] chosen for this task. MASSOUD (Multidisciplinary

Aero/Struc Shape Optimization Using Deformation is a geometry parameterization tool that utilizes soft object animation algorithms used in computer graphics. It parameterizes the shape perturbations rather than the geometry itself, and relates grid deformation to aerodynamics shape design variables such as thickness, camber, twist, shear, and planform. The morphing capabilities available through MASSOUD are independent of grid topology, making it suitable for a variety of analysis codes such as CFD and CSM. Sensitivity derivatives are available in this software, and can be used for gradient-guided optimization. This algorithm is suitable for both low-fidelity (e.g., linear aerodynamics and equivalent laminated plate structures) and high-fidelity analysis tools (e.g., nonlinear CFD and detailed CSM modeling) [20]. A schematic of the data flow in this BB can be seen in Figure 46 below.

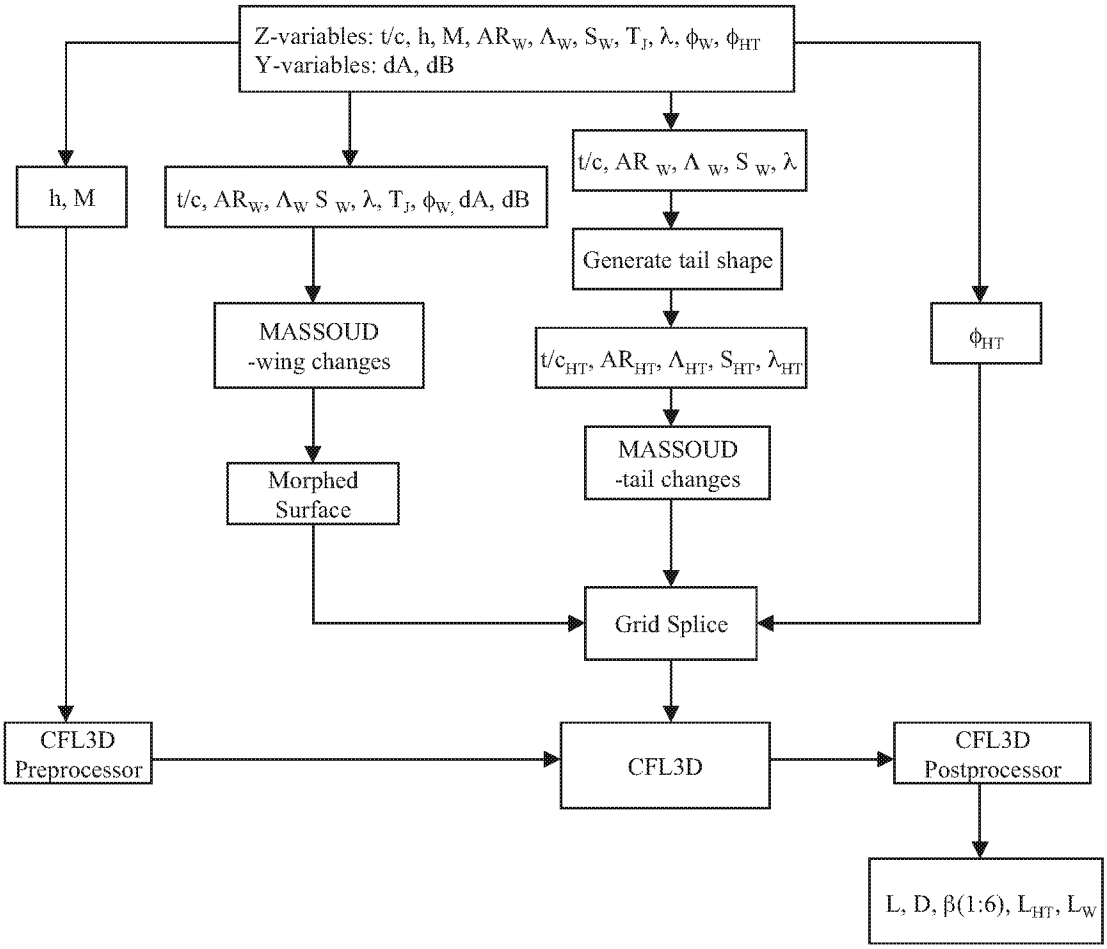


**Figure 46. System Variable Flow for New BLISS-RS Aero BB**

Figure 46 also shows the output values for $L_{HT}$ and $L_W$, lift of the horizontal tail and wing, respectively. These values are used in a trim constraint during system optimization. Previous versions of BLISS performed this operation within the aerodynamics BB.

## Improved Structures Black Box

In the upgraded version of BLISS, the actual lift distribution is passed along to the structures BB from the Aero BB via CFD analysis. The six coefficients $\beta$ are used to describe the quadratic representation of the lift distribution. This is an improvement to the previous assumed elliptical /trapezoidal (Shrenk Approximation) distribution, thus the

accuracy of this distribution is greatly enhanced. Figure 47 shows the system variable flow for the improved structures BB. The notion of twist used in previous discussions has been abandoned here in favor of dA and dB, the vertical displacement at the wing tip leading edge and trailing edge, respectively.

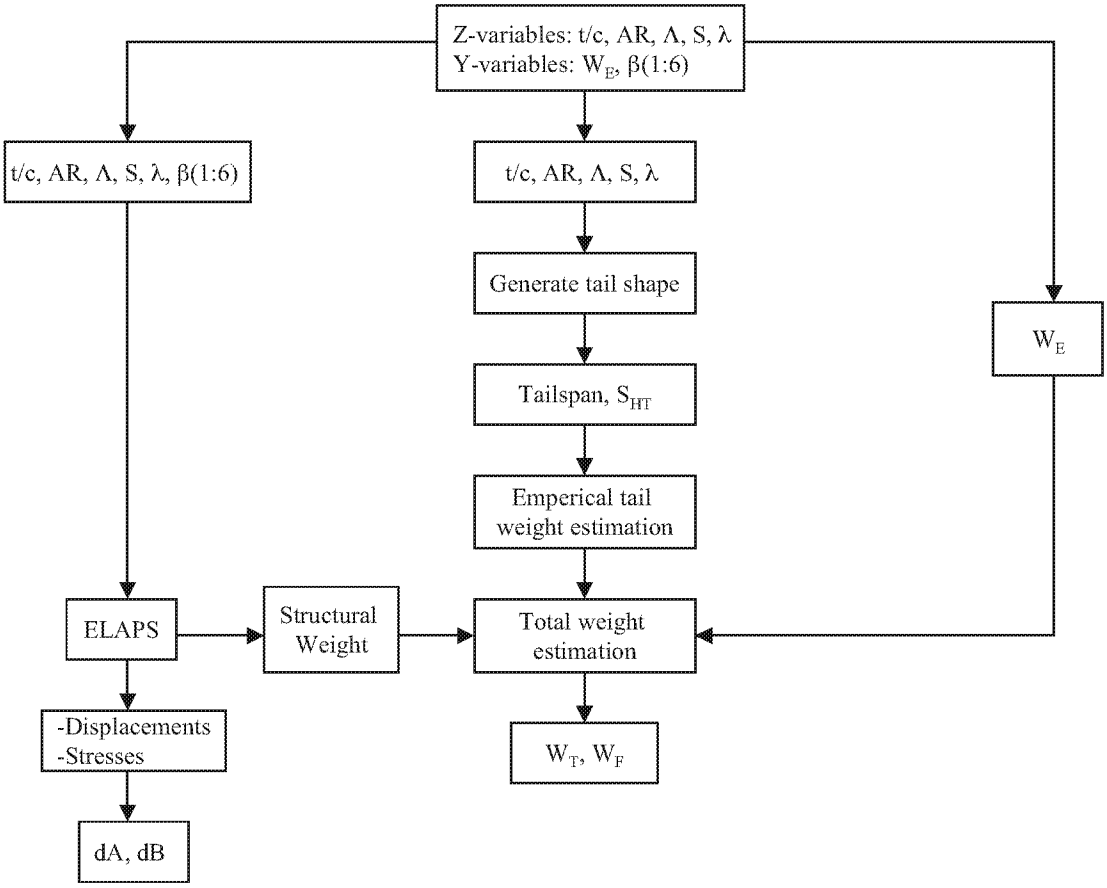


**Figure 47. System Variable Flow for New BLISS-RS structures BB**

Another important modification to the structures BB is the implementation of an anti-flutter constraint during local optimization. The initial formulation of the flutter constraint (explained in detail in Appendix B) is based on the ratio of deformed wing bending moment, $M_{DR}$, to the rigid wing root bending moment, $M_{RR}$. However, a higher-fidelity flutter calculation is possible using ELAPS.

The changes to the aerodynamics and structures modules have an effect on the overall data flow of the system. Figure 48 shows the new data flow associated with implementation of the new aerodynamics and structures BB's. Likewise, Figure 49 diagrams the system optimization process associated with the new BB's.

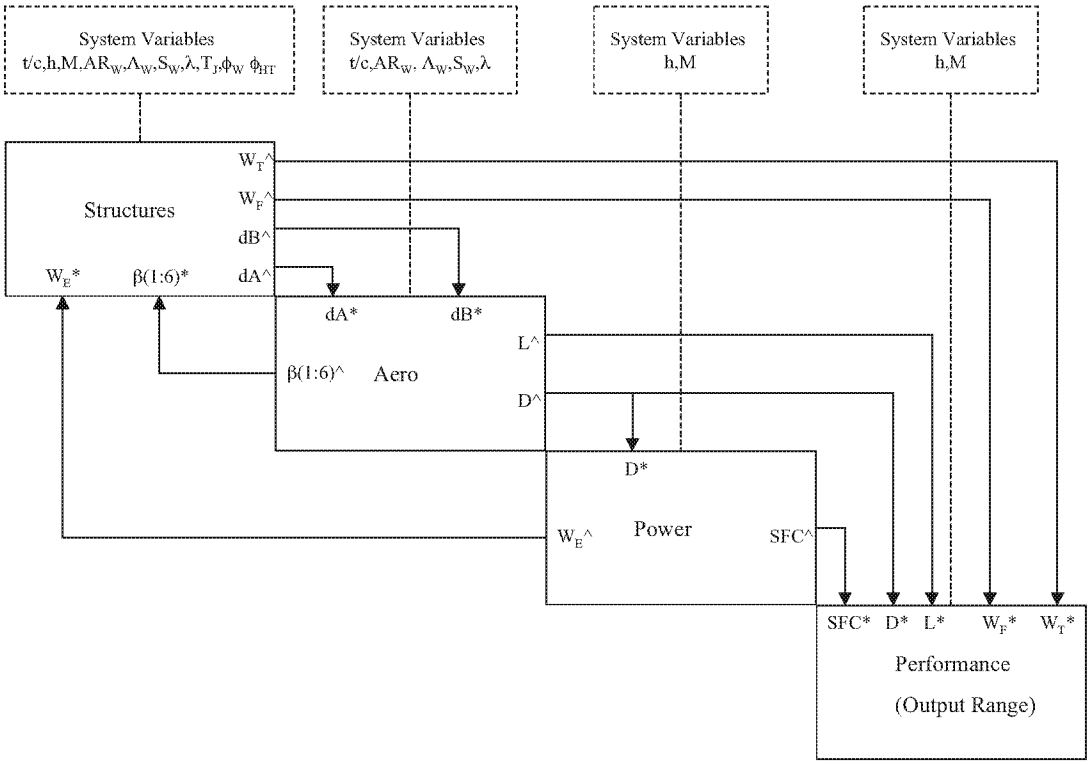


**Figure 48. New BLISS-RS System Data Flow**

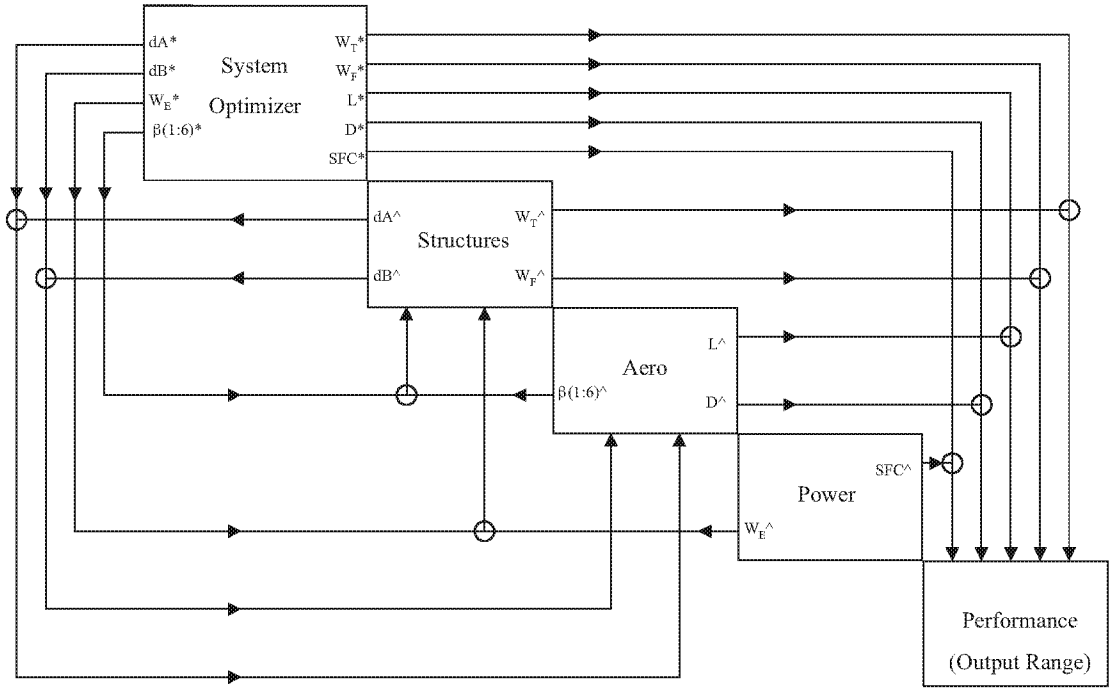


**Figure 49. New BLISS RS System Optimizer**

# 9. Conclusions and Future Work

The results presented herein reflect a change in the overall BLISS process: a shift from gradient-guided optimization methods to RSM based schemes. BLISS-RS allows autonomous, distributed, and concurrent optimization in subsystems (disciplines) off-line. The method decouples local variables and constraints from the system level variables and coupling constraints, and optimizes the system for its objective and satisfies couplings. The response surface methodology was successfully implemented and tested on a conceptual aircraft design. By demonstrating the ability to use response surfaces in the BLISS-RS procedure, a number of possible improvements and follow-up work have emerged.

First, the logical next step in the evolution of BLISS would be study its performance in a parallel-processing environment. The real gains to be made through

response surface methods are offset by the computational labor involved to create the RS's. Since the RS's generated in the current research were done serially, these gains were difficult to assess.

The modular nature of BLISS-RS easily allows for improvements to the analyses local to each BB. Indeed, improvement in the fidelity of the black boxes was demonstrated in BLISS-98 [1-4], and again in BLISS-RS with the improved aerodynamics BB. A possible extension of this idea would be to increase the number of BB's, therefore potentially increasing the number of all levels of design variables. Prospective new BB's could include a stability and control analysis or even cost analysis. The process could also be adapted to reflect a more realistic design, for example by considering the entire flight envelope, or by more specific modeling. Therefore, BLISS-RS could be used in more advanced stages of the design process.

Throughout the course of the research, many trial and error based decisions were conducted so there remain many possibilities for further investigation into "the road not taken". The choice of point placement schemes was based primarily on judgmental criteria. Therefore, a definitive study should be conducted to find the best scheme for accurate results, and one that is practical to implement. The predominant case in this regard is the manner in which response surfaces are created from one iteration to the next. The BLISS-RS process can be made more efficient by using data points from previous iterations circumventing the need to produce a full set of points each iteration. This problem was addressed in the current research, but the dimensionality of the problem limited the number of points that could be retained, while still giving a set of points that fell within the interval of the next iteration.

Lastly, the user interface of BLISS is a prime candidate for improvement. As the program stands, user defined inputs are inserted directly into the source code. This presents many challenges (experienced firsthand by the author) when trying to make small changes in baseline aircraft configuration, BB analyses, termination tolerances, etc. It will be essential that the interface becomes more user-friendly if BLISS-RS is to be used in actual design applications.

# References


1 Rao, S. S.: *Engineering Optimization: Theory and Practice,* 3rd ed., New York: Wiley, 1996.

2. Sobieszczanski-Sobieski, J.; Agte, J.; and Sandusky, R.: "Bi-Level Integrated System Synthesis," Proc. 7th AIAA/USA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper No. 98-4916, Sept. 1998.

3. Sobieszczanski-Sobieski, J.; Agte, J.; and Sandusky, R.: "Bi-Level Integrated System Synthesis (BLISSS)", NASA/TM-1998-208715, August, 1998.

4. Sobieszczanski-Sobieski, J.; Emiley, M; Agte, J.; and Sandusky, R.: "Advancement of Bi-Level Integrated System Synthesis (BLISSS)", Proc. 38th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper No. 2000-0421, January, 2000.

5. Raymer, D. P.: *Aircraft Design: A Conceptual Approach*, AIAA Educational Series, 1992.

6. Giles, G.L.: *ELAPS User's Manual (Preliminary Draft)*, NASA Langley Research Center, Hampton, VA, October 1997.

7. Giles, G.L.: "Equivalent Plate Modeling for Conceptual Design of Aircraft Wing Structures", Proc. 1st AIAA Aircraft Engineering Technology and Operations Congress, Los Angeles, CA, AIAA Paper No. 95-3945, September 1995.

8. Giles, G.L.: "Further Generalization of an Equivalent Representation for Aircraft Structural Analysis", *Journal of Aircraft,* Vol. 26, No.1, January 1989, pp. 67-74.

9. Giles, G.L.: "Equivalent Plate Analysis of Aircraft Wing Box Structures with General Planform Geometry", *Journal of Aircraft*, Vol. 23, No. 11, November 1986, pp. 859-864.

10. Sextone, M. G.: "Aircraft Structural Mass Property Prediction Using Conceptual-Level Structural Analysis", Proc. 57th SAWE Annual Conference, Wichita, Kansas, SAWE Paper No. 2410, May, 1998.

11. Sobieszczanski-Sobieski, J.: Unpublished Correspondence, NASA Langley Research Center, 2001.

12. Myers, R. H. and Montgomery, D.C.: *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, New York: Wiley, 1995.

13. Unal, R.; Braun, R. D.; Moore, A. A.; Lepsch, R. A.: "Response Surface Model Building Using Orthogonal Arrays for Computer Experiments", 19th Annual International Conference of the International Society of Parametric Analysis, New Orleans, LA, May 27-30, 1997.

14. Loretti, A. J.: "Generation of Response Surfaces for Aircraft Structural Wing Weight with Strength and Flutter Sizing", GWU MS Thesis, August, 1997.

15. Knuth, D. E.: *The Art of Computer Programming*, Vol. 2, pp.103-104, Reading, MA: Addison-Wesley, 1997.

16. Venter, G.: Unpublished Correspondence, Vanderplaats Research and Development, Inc., 2001.

17. MATLAB Manual, Version 6.1, Release 12.1, the MathWorks, Inc., May, 2001.

18. Venter, G. and Watson, B.: "Efficient Optimization Algorithms for Parallel Applications", Proc. Of 8th AIAA/USAF/NASA/ISSMO Symposium at Multidisciplinary Analysis and Optimization, Long Beach, CA, September, 2000.

19. Rumsey, C.L.; Biedron, R.T.; and Thomas, J.L.: CFL3D: Its History and Some Recent Applications, NASA TM-112861, NASA Langley Research Center, Hampton, VA 23681-0001, May 1998.

20. Samareh, J.A.: "Novel Multidisciplinary Shape Parameterization Approach", Journal of Aircraft (American Institute of Aeronautics and Astronautics), Vol. 38, No. 6, 2001, pp. 1015-1024.

# Appendix A. Hypersphere Point Placement

This method generates uniformly distributed random points on the surface of the hypersphere, then employs an antibunching mechanism. A brief synopsis of the algorithm follows from [11] and [15]:

1). Generate normally distributed deviates $X_1, X_2, \ldots, X_n$ with mean zero and variance one.

2). Obtain coordinates of points on the surface of the sphere by dividing each deviate by the root sum of the squares of the deviates:

$$\left(\frac{X_1}{r}, \frac{X_2}{r}, \ldots, \frac{X_n}{r}\right) \text{ where } r = \sqrt{X_1^2 + X_2^2 + \ldots + X_n^2}. \qquad \textbf{Equation A1}$$

Since the distribution function for the point $(X_1, X_2, \ldots, X_n)$ has a density that only depends on the distance from the origin, it has uniform distribution when projected onto the surface of the sphere.

To protect against the possibility of a point falling very closely to or on top of its nearest neighbor, an "anti-bunching" mechanism was developed. Overlapping points would in effect act like a single point, therefore reducing the total number of points by one. Since the goal here is to generate the fewest points required to fit a quadratic surface, an overlapping set of points would drop the number of points below the minimum (NS). Additionally, the accuracy of the fitted surface is reduced when the points are bunched together.

The solution for this problem is achieved by checking the distance of a point generated to its nearest neighbor. The distance between two points, A and B can be shown as

$$dist_{AB} = \sqrt{\sum_{i=1}^{NX} (XA_i - XB_i)} \qquad \textbf{Equation A2}$$

where $XA$ and $XB$ are the vectors containing the NX-coordinates of points A and B, respectively. If the points falls too close to previous points, it is rejected and another point is generated until one meets a certain criteria.

To introduce the notion of an anti-bunching criterion, suppose an N-dimensional coordinate system was partitioned into intervals (lower bound through upper bound) on each axis. After normalizing the lengths of the intervals to one, each is then divided into D divisions of equal length,

$$S = \frac{1}{D} \qquad \textbf{Equation A3}$$

The quantity S can be thought of as a measure of distance from a point to its nearest neighbor expressed in units of the normalized interval. It may also be regarded as a measure of the mesh density. If each division on every axis were sampled at its center, the number of points generated would be $NP = D^N$. After inverting this expression, the corresponding mesh density is

$$S = NP^{-\frac{1}{N}}. \qquad \textbf{Equation A4}$$

To prevent bunching, the distance between points can be limited to a 1/h fraction of S, or

$$dist_{AB} \geq \frac{S}{h} \qquad \textbf{Equation A5}$$

Where h is a user-defined input, the anti-bunching factor. From this definition, a high anti-bunching factor will result in more bunching of points whereas lowering the value of h makes the point dispersion approach an even distribution, as shown in figures A1 through A4. Also note, a single center point has been generated at the origin in each case.

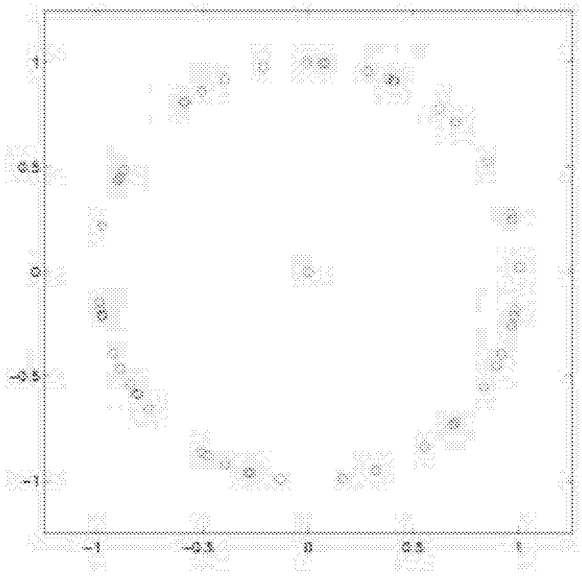

**Figure A1. Distribution of 50 points on a two-dimensional hypersphere with h=100.**

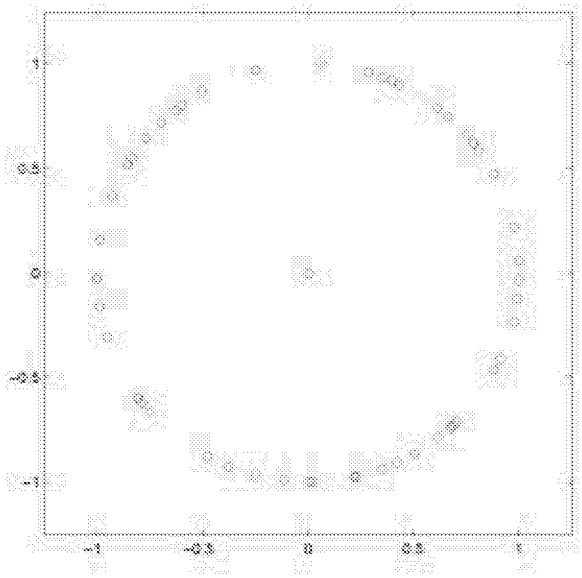

**Figure A2. Distribution of 50 points on a two-dimensional hypersphere with h=10.**

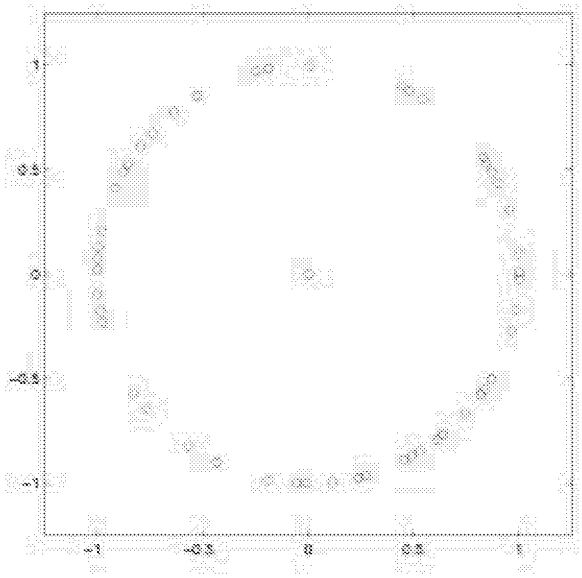

**Figure A3. Distribution of 50 points on a two-dimensional hypersphere with h=5.**

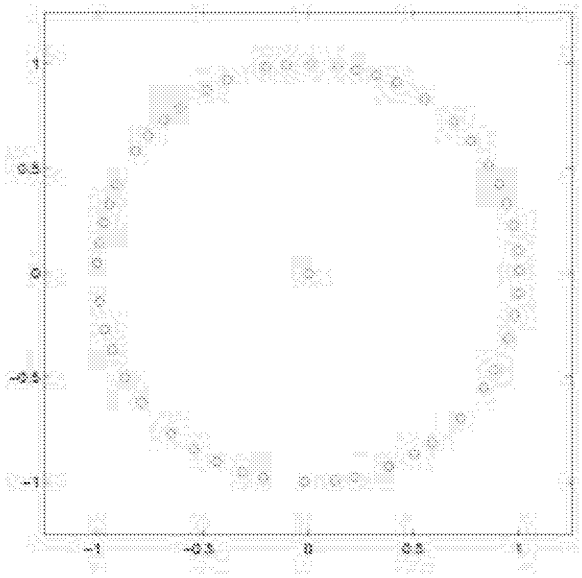

**Figure A4. Distribution of 50 points on a two-dimensional hypersphere with h=1.5.**

As expected, a lower anti-bunching factor results in a more uniform distribution. It is important to note that the anti-bunching factor must not fall below the value that gives a truly even distribution. If this were the case, all subsequent points generated would violate the distance criterion of equation A5. For the case set of 50 points given above (49 on the surface and one center run), this critical value is $h_{crit}$=1.112.

The volume of a hypersphere can be found as from equation A6:

$$Volume = \frac{\pi^{\frac{n}{2}}}{\Gamma\left[\frac{n}{2}+1\right]} r^n \qquad \textbf{Equation A6}$$

where $n$ is the number of independent variables, and

$$\Gamma(n) = \int_0^\infty x^{n-1} e^{-x} dx \qquad \textbf{Equation A7}$$

or

$$\Gamma(n) = \frac{\pi^{\frac{n}{2}}}{\left(\frac{n}{2}\right)!} r^n \text{ if } n \text{ even} \qquad \textbf{Equation A8}$$

$$\Gamma(n) = \frac{\pi^{\frac{n}{2}}}{\left(\frac{n}{2}\right)!} r^n \text{ if } n \text{ odd.} \qquad \textbf{Equation A9}$$

# Appendix B. Empirical Flutter Constraint Formulation

The following formulation of an empirical flutter constraint was adapted from [11]. The method limits excessive flutter by comparing the root bending moment of a twisted wing to that of a rigid wing. A more realistic formulation that includes vibration frequency of the wing is also included.

To introduce the notion of a twist constraint, consider the change in local angle of attack, $\Delta\alpha$, of a spanwise station along the wing:

$$\Delta\alpha = \frac{(dA - dB)}{c_{AB}} \qquad \textbf{Equation B1}$$

where $dA$ and $dB$ are the z-displacements of points A and B (leading and trailing edge locations, respectively) and $c_{AB}$ is the chord length measured from point A to point B. The aerodynamics module uses $\Delta\alpha$ to modify the wing shape and to compute the resulting changes in aerodynamic loads. Suppose now that the pressure distribution along the span of the wing is given, either by assuming a distribution (as was done in BLISS 98) or by computing the distribution via CFD. The total pressure distribution is given by

$$F_{av}(x) = a \cdot f(x) \qquad \textbf{Equation B2}$$

where $f(x)$ is the pressure distribution, and $a$ is a constant that controls the lift magnitude. However, the lift given by equation B2 is valid only for rigid wings. Therefore, we must account for wing bending and twisting so that a chord at location x rotates per equation

B1. This rotation results in altering the local angle of attack. The effect of $\Delta\alpha$ on total lift is

$$\Delta L(x) = a \cdot f(x)\frac{\Delta\alpha(x)}{\alpha_r} \qquad \textbf{Equation B3}$$

where $\Delta\alpha$ is given for the local chord in equation B3 and $\alpha_r$ is the angle of attack of the root chord. Now the total lift can be determined as an integrated sum of the pressure distribution and the change in lift:

$$L = a \cdot \int f(x)\left[C(x) + \frac{\Delta\alpha}{\alpha_r}\right]dx \qquad \textbf{Equation B4}$$

where C(x) represents the wing geometry (root chord, tip chord, and span for a trapezoidal wing). In equation B4, the integration extends from root to tip. This integrated pressure must match the required lift $L_0$ input to the structures BB from the system optimizer so that

$$a = \frac{L_0}{\int f(x)\left[C(x) + \frac{\Delta\alpha}{\alpha_r}\right]dx} \qquad \textbf{Equation B5}$$

A trade off between structural weight and drag occurs because the wing loses some angle of attack to $\Delta\alpha$. The loss in angle of attack intensifies outboard and the related loss in lift ($\Delta$L in equation B3) results in a decrease in wing bending moment since the center of lift moves inboard. The reduction of wing bending moment is an advantage in terms of structural weight, but results in increased aerodynamic drag since the lift distribution departs from the minimum drag, elliptical distribution. However, for a transport aircraft, the drag penalty can be ignored if the wing is built to a jig shape that incorporates the negative twist anticipated at 1 g flight. In 1 g flight, the wing flexes to

an ideal shape as if it were rigid, but still relieves the bending load in a pull-up maneuver. Hence, we can exploit the wing flexing as a natural mechanism to reduce the pull-up bending and thus save weight to improve range.

The deformed wing root bending moment can be defined as

$$M_{DR} = a \cdot \int f(x) \cdot \left[ C(x) + \frac{\Delta\alpha}{\alpha_r} \right] x dx \qquad \textbf{Equation B6}$$

Similarly, a rigid wing root bending moment can be written as

$$M_{RR} = a \cdot \int f(x) \cdot C(x) x dx \qquad \textbf{Equation B7}$$

Since $\Delta\alpha$ <0 for an aft-swept wing, $r$ can be introduced as

$$r = \frac{M_{DR}}{M_{RR}} < 1 \qquad \textbf{Equation B8}$$

When the value of r is less than unity, the wing is relieved of the bending load. For local optimization, r is used as a constraint to prevent excessive flutter. For example, the simplest form of the constraint would be a bound on r:

$$r < \tilde{r} \qquad \textbf{Equation B9}$$

where $\tilde{r}$ could be a values such as 0.9, it which case 10% bending relief would be allowed. A better approximation can be formulated based on the formulas of vibration frequency. The formulas for bending and torsional frequency are given in equations B10 and B11, respectively.

$$\omega_b = P_1 \sqrt{\frac{k_b}{m}} \qquad \textbf{Equation B10}$$

$$\omega_t = P_2\sqrt{\frac{k_t}{J}}$$ **Equation B11**

where $P_1$ and $P_2$ are constants of proportionality, $k_b$ is the bending stiffness corresponding to the wing tip unit torque, $k_t$ is the torsion stiffness corresponding to the wing tip unit torque, m is wing structural mass, and J is the wing moment of inertia associated with rotation about the elastic axis. J can be expressed in terms of m and radius of gyration, R, by:

$$J = mR^2$$ **Equation B12**

If the wing is free of flutter at some reference state (designated with a ~), limits can be set around $\omega_b$ and $\omega_t$ ($u_b$ and $u_t$) so that

$$g = \frac{\omega_b}{\tilde{\omega}_b} - (1 - u_b) < 0$$ **Equation B13**

$$g = (1 - u_b) - \frac{\omega_b}{\tilde{\omega}_b} < 0$$ **Equation B14**

These constraints keep $\omega_b$ within $\pm u_b$ of $\tilde{\omega}_b$. The actual value of $u_b$ is somewhat arbitrary, for example, 0.1. Similar constraints can be generated for $\omega_t$. The reference state ~ can be determined by executing BLISS with the constraint in equation B9. Then the constraint given in equation B9 may be replaced with equations B13 and B14. Note that in equations B13 and B14, $m_s$, $k_b$, and $k_t$ are the only items that need to be computed since $P_1$, $P_2$, and R cancel out (under a simplifying assumption that they remain approximately constant).

# Appendix C. Data Tables

| | Iteration Number | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Variable | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| t1(inner) | 2.00 | 1.9978 | 1.9952 | 1.9948 | 1.9976 | 1.9955 | 1.9947 | 1.9946 | 1.9942 | 1.9939 | 1.9937 |
| t1(middle) | 2.00 | 1.9978 | 1.9952 | 1.9948 | 1.9976 | 1.9955 | 1.9947 | 1.9946 | 1.9942 | 1.9939 | 1.9937 |
| t1(outer) | 2.00 | 1.9978 | 1.9952 | 1.9948 | 1.9976 | 1.9955 | 1.9947 | 1.9946 | 1.9942 | 1.9939 | 1.9937 |
| t2(inner) | 2.00 | 1.9998 | 1.9995 | 1.9997 | 1.9999 | 1.9998 | 1.9998 | 1.9998 | 1.9998 | 1.9998 | 1.9998 |
| t2(middle) | 2.00 | 1.9998 | 1.9995 | 1.9997 | 1.9999 | 1.9998 | 1.9998 | 1.9998 | 1.9998 | 1.9998 | 1.9998 |
| t2(outer) | 2.00 | 1.9998 | 1.9995 | 1.9997 | 1.9999 | 1.9998 | 1.9998 | 1.9998 | 1.9998 | 1.9998 | 1.9998 |
| t3(inner) | 2.00 | 1.9978 | 1.9952 | 1.9948 | 1.9976 | 1.9955 | 1.9947 | 1.9946 | 1.9942 | 1.9939 | 1.9937 |
| t3(middle) | 2.00 | 1.9978 | 1.9952 | 1.9948 | 1.9976 | 1.9955 | 1.9947 | 1.9946 | 1.9942 | 1.9939 | 1.9937 |
| t3(outer) | 2.00 | 1.9978 | 1.9952 | 1.9948 | 1.9976 | 1.9955 | 1.9947 | 1.9946 | 1.9942 | 1.9939 | 1.9937 |
| ts1(inner) | 4.00 | 3.9703 | 4.0242 | 4.1032 | 3.9814 | 3.8414 | 3.7784 | 3.8343 | 3.8060 | 3.8197 | 3.8404 |
| ts1(middle) | 4.00 | 3.9760 | 4.0170 | 4.0716 | 3.9867 | 3.8884 | 3.8442 | 3.8833 | 3.8634 | 3.8730 | 3.8867 |
| ts1(outer) | 4.00 | 3.9818 | 4.0097 | 4.0399 | 3.9920 | 3.9354 | 3.9100 | 3.9323 | 3.9209 | 3.9263 | 3.9330 |
| ts2(inner) | 4.00 | 3.9999 | 3.9998 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 |
| ts2(middle) | 4.00 | 3.9999 | 3.9998 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 |
| ts2(outer) | 4.00 | 3.9999 | 3.9998 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 | 3.9999 |
| ts3(inner) | 4.00 | 3.9703 | 4.0242 | 4.1032 | 3.9814 | 3.8414 | 3.7784 | 3.8343 | 3.8060 | 3.8197 | 3.8404 |
| ts3(middle) | 4.00 | 3.9760 | 4.0170 | 4.0716 | 3.9867 | 3.8884 | 3.8442 | 3.8833 | 3.8634 | 3.8730 | 3.8867 |
| ts3(outer) | 4.00 | 3.9818 | 4.0097 | 4.0399 | 3.9920 | 3.9354 | 3.9100 | 3.9323 | 3.9209 | 3.9263 | 3.9330 |
| Λht (deg) | 60 | 70.00 | 70.00 | 70.00 | 70.00 | 62.19 | 70.00 | 67.36 | 70.00 | 70.00 | 70.00 |
| Lw (%MAC) | 10 | 1.00 | 5.69 | 5.68 | 1.00 | 1.00 | 4.33 | 2.75 | 4.33 | 4.33 | 1.00 |
| Lht (%MAC) | 250 | 212.64 | 350.00 | 350.00 | 211.53 | 266.82 | 350.00 | 350.00 | 350.00 | 350.00 | 350.00 |
| T (%) | 35 | 21.17 | 31.39 | 25.22 | 22.76 | 21.97 | 20.76 | 20.21 | 20.69 | 20.64 | 20.62 |

**Figure C1. BLISS-RS Results for Local Variables**

| | Iteration Number | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Variable | Initial | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| t/c | 0.08 | 0.10 | 0.10 | 0.06 | 0.06 | 0.05 | 0.05 | 0.04 | 0.04 | 0.04 | 0.03 |
| h (ft) | 55000 | 60000.00 | 54946.99 | 60000.00 | 60000.00 | 60000.00 | 60000.00 | 60000.00 | 60000.00 | 59650.05 | 60000.00 |
| M | 1.8 | 1.69 | 1.78 | 1.83 | 1.75 | 1.72 | 1.67 | 1.65 | 1.67 | 1.65 | 1.66 |
| AR wing | 4 | 3.50 | 3.50 | 3.50 | 3.50 | 3.50 | 3.50 | 3.50 | 3.57 | 3.50 | 3.73 |
| Λw (deg) | 45 | 40.00 | 40.00 | 40.00 | 40.00 | 40.00 | 40.00 | 40.00 | 40.75 | 40.53 | 40.00 |
| Sref (ft^2) | 400 | 200.00 | 446.58 | 481.69 | 442.72 | 418.50 | 489.69 | 495.28 | 534.80 | 564.84 | 578.08 |
| Sht (ft^2) | 120 | 95.64 | 69.76 | 83.31 | 101.24 | 114.69 | 124.77 | 109.87 | 115.54 | 111.29 | 108.10 |
| AR ht | 4.5 | 4.27 | 4.69 | 4.23 | 4.65 | 4.96 | 5.07 | 4.89 | 5.02 | 4.92 | 4.85 |
| taper ratio | 0.2 | 0.40 | 0.18 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.11 |

**Figure C2. BLISS-RS Results for System Variables**

| | Iteration Number | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Variable | Initial | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Wt (lb) | 33278.12 | 34767.72 | 37277.26 | 35120.17 | 33034.16 | 31994.33 | 31974.33 | 31728.41 | 31508.81 | 31436.23 | 31342.34 |
| Wf (lb) | 15109.47 | 18886.84 | 22664.21 | 19587.40 | 17274.85 | 16574.57 | 16707.70 | 16376.87 | 16046.05 | 16000.85 | 15888.08 |
| $C_D$ (ft^2) | 6.92 | 6.40 | 6.28 | 6.55 | 6.28 | 6.41 | 6.34 | 6.32 | 6.29 | 6.32 | 6.30 |
| L (lb) | 33278.12 | 34767.75 | 37277.22 | 35120.13 | 33034.19 | 31994.30 | 31974.30 | 31728.38 | 31508.78 | 31436.19 | 31342.37 |
| D (lb) | 8701.50 | 7272.81 | 5982.93 | 4693.05 | 4256.50 | 3780.13 | 3383.76 | 3495.47 | 3462.84 | 3488.63 | 3486.77 |
| L/D | 3.82 | 4.78 | 5.74 | 7.48 | 5.96 | 7.40 | 7.86 | 8.23 | 8.27 | 8.39 | 8.51 |
| SFC | 1.15 | 1.01 | 1.15 | 1.14 | 1.09 | 1.11 | 1.10 | 1.08 | 1.10 | 1.09 | 1.09 |
| We (lb) | 7284.91 | 5463.68 | 3793.06 | 4643.03 | 4866.27 | 4533.20 | 4258.10 | 4382.89 | 4447.87 | 4384.30 | 4397.49 |
| ESF | 1.00 | 0.63 | 0.47 | 0.54 | 0.57 | 0.53 | 0.50 | 0.51 | 0.53 | 0.52 | 0.52 |

**Figure C3. BLISS-RS Results for Coupling Variables**

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| 05-2002 | Contractor Report | |

**4. TITLE AND SUBTITLE**
A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS)

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Troy David Altus

**5d. PROJECT NUMBER**
NCC1-387

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
706-25-02-14

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
George Washington University
Joint Institute for Advancement of Flight Sciences
Hampton, Virginia
NASA Langley Research Center
Hampton, VA 23681-2199

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
National Aeronautics and Space Administration
Washington, DC 20546-0001

**10. SPONSORING/MONITOR'S ACRONYM(S)**
NASA

**11. SPONSORING/MONITORING REPORT NUMBER**
NASA/CR-2002-211652

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified - Unlimited
Subject Category 05
Availability: NASA CASI (301) 621-0390 Distribution: Nonstandard

**13. SUPPLEMENTARY NOTES**
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://techreports.larc.nasa.gov/cgi-bin/NTRS
Langley Technical Monitor: Jaroslaw Sobieski

**14. ABSTRACT**

The report describes a new method for optimization of engineering systems such as aerospace vehicles whose design must harmonize a number of subsystems and various physical phenomena, each represented by a separate computer code, e.g., aerodynamics, structures, propulsion, performance, etc. To represent the system internal couplings, the codes receive output from other codes as part of their inputs. The system analysis and optimizati task is decomposed into subtasks that can be executed concurrently, each subtask conducted using local state and design variables and holding consta a set of the system-level design variables. The subtasks results are stored in form of the Response Surfaces (RS) fitted in the space of the system-leve variables to be used as the subtask surrogates in a system-level optimization whose purpose is to optimize the system objective(s) and to reconcile the system internal couplings. By virtue of decomposition and execution concurrency, the method enables a broad workfront in organization of an engineering project involving a number of specialty groups that might be geographically dispersed, and it exploits the contemporary computing technology of massively concurrent and distributed processing. The report includes a demonstration test case of supersonic business jet design.

**15. SUBJECT TERMS**
Optimization by Decomposition, Engineering System Design, Parallel Processing

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | STI Help Desk (email: help@sti.nasa.gov) |
| | | | | | 19b. TELEPHONE NUMBER *(Include area code)* |
| U | U | U | UU | 90 | (301) 621-0390 |