AIAA JOURNAL
Vol. 38, No. 8, August 2000

# Bilevel Integrated System Synthesis with Response Surfaces

Srinivas Kodiyalam*
*Lockheed Martin Space Systems Company, Sunnyvale, California 94089*
and
Jaroslaw Sobieszczanski-Sobieski†
*NASA Langley Research Center, Hampton, Virginia 23681*

**Two variants of the BLISS method are proposed for multidisciplinary design optimization (MDO). These variants utilize experimental design methods and response surfaces to reduce the number of expensive system analysis required for solution of the MDO problem. BLISS is an MDO method for decomposition-based optimization of engineering systems that involves system optimization with a relatively small number of design variables and a number of subsystem optimizations that could each have a large number of local variables. In BLISS, the optimum sensitivity analysis data are used to relate the subsystem optimization solutions with the system optimizations. Instead, with the proposed variants, polynomial response surface approximations using either the system analysis or the subsystem optimization results are used. Additionally, the response surface construction process is well suited for computing in a concurrent processing environment. The proposed variants are implemented and evaluated on a conceptual-level aircraft and ship design problem.**

## I. Introduction

THE design of complex systems, such as aerospace, automotive, and manufacturing processes, are composed of multiple subsystems with complex interactions, where stringent and often conflicting design requirements are imposed. To solve such design and optimization problems effectively, it is essential that these systems be decomposed into more manageable disciplinary problems that can be concurrently solved while simultaneously accounting for the interactions among the different disciplines.[1] A number of multidisciplinary design optimization (MDO) methods have been formulated in recent years and a survey of these methods is provided in Refs. 2 and 3. Although the architecture of some of these MDO methods may not be entirely intuitive, their solution approach provides for a much more practical and efficient path to reach an optimal solution over the conventional all-in-one approach.

The all-in-one (A-i-O) formulation (referred to as multidisciplinary feasibility in Ref. 4) is the conventional approach for solving the MDO problem. Although the method is well understood, the principal drawbacks are its expense in having to perform a complete multidisciplinary analysis at every stage, including computing derivatives and its lack of modularity and fit with respect to integrated product teams in industry. The concurrent subspace optimization (CSSO) method[1,5,6] provides for multidisciplinary feasibility at each CSSO cycle and separate optimizations within the subsystems but deals with all of the design variables simultaneously at the system/coordination problem level. The latter approaches to CSSO make use of response surfaces at the system level that is not effective for design spaces of dimensionality over approximately 20. The collaborative optimization (CO) method[7,8] and individual discipline feasible (IDF) method[4] dispenses with multidisciplinary compatibility at each system iteration; instead the compatibility is attained as the system optimization problem converges. CO has the appealing property of disciplinary autonomy that fits well with the industry settings, but has so far been demonstrated only for problems where the interdisciplinary coupling has a small bandwidth. A large number of coupling variables between the disciplines would proportionally increase the number of system-level optimization variables in CO.

The recently introduced BLISS method[9] uses a gradient-guided path to reach the improved system design, alternating between the set of modular design subspaces (disciplinary problems) and the system-level design space. BLISS is an A-i-O-like method because a complete system analysis is performed to maintain multidisciplinary feasibility at the beginning of each cycle of the path. However, the system-level optimization problem with BLISS uses a relatively small number of design variables that are shared by the subspaces (disciplines). In addition, the solution of the system-level problem is obtained using the derivatives of the behavior (state) variables with respect to system-level design variables and the Lagrange multipliers of the constraints obtained at the solution of the disciplinary optimizations.

This paper explores the use of response surfaces[10] for solution of the system-level optimization problem in BLISS. Its objectives are replacing the need for subsystems' optimum sensitivity analysis, eliminating the need for subsystem optimizations to yield a feasible solution for each BLISS cycle, and reducing the total work, where total work is computed as the total number of disciplinary (black box) analyses, including those required during system analysis, sensitivity analysis, and disciplinary optimizations, required for convergence of the BLISS procedure. The limitation, however, with the use of response surfaces is that the number of independent system-level design variables will be restricted to approximately 25 variables.

## II. Optimization Problem and BLISS Procedure

The general system optimization problem is stated in the following form. Given a set of design variables, $X$,

$$\begin{aligned} &\text{Find:} && \Delta X \\ &\text{Minimize:} && \Phi[X, Y(X)] \\ &\text{Satisfy:} && G[X, Y(X)] \end{aligned} \tag{1}$$

In the problem defined by Eq. (1), $Y(X)$ represents the behavior (state) variables, $\Phi$ represents the design objective function, and $G$ represents the design constraints.

With BLISS, the general system optimization problem is decomposed into a set of local optimizations dealing with a large number of detailed local design variables ($X$) and a system-level optimization dealing with a small number of global variables ($Z$). Optimum sensitivity data are used to link the subsystem (local) and system-level optimizations. The details of the complete BLISS procedure is provided in Ref. 11. For completeness, the key steps in the BLISS procedure are outlined next.

---

Received 2 August 1999; revision received 17 January 2000; accepted for publication 18 January 2000. 

*Staff Engineer, Advanced Technology Center, 1111 Lockheed Martin Way; srinivas.kodiyalam@lmco.com. Senior Member AIAA.

†Manager, Computational AeroSciences and Multidisciplinary Research Coordinator; j.sobieski@larc.nasa.gov. Fellow AIAA.

1) Initialize local disciplinary ($X$) and system-level ($Z$) variables.

2) Perform system analysis (SA) to compute the state variables $Y$ and the design constraint functions $G$; the SA includes all of the local disciplinary analysis or herein referred to as black box analysis (BBA) for all the black boxes/disciplines (BB).

3) Check for convergence of the BLISS procedure.

4) Perform black box sensitivity analysis to compute local derivatives, including d($Y$, $X$), d($Y_{r,s}$, $Y_r$), d($G$, $Z$), and d($G$, $Y$). Perform system sensitivity analysis to compute global derivatives D($Y$, $X$) and D($Y$, $Z$). Note that the subscript $r$ refers to the $r$th BB/discipline, $Y_r$ corresponds to the vector of state variables output from BB$_r$, and $Y_{r,s}$ correspond to vector of variables input to BB$_r$ from BB$_s$.

5) Perform black box optimization (BBOPT) for all of the BBs to get the change in local design variables $\Delta X_{\text{opt}}$ and the Lagrange multipliers $L$ for the active constraints at the constrained optimum. The local optimization problem for a representative BB$_k$ can be stated in the following form:

$$\begin{aligned} &\text{Given:} && X_k,\ Z, \text{ and } Y_{k,m} \\ &\text{Find:} && \Delta X_k \\ &\text{Minimize:} && \phi_k = \mathrm{D}(y_{r,i}, X_k)^T \Delta X_k \\ &\text{Satisfy:} && G_k \le 0, \text{ including side constraints} \end{aligned} \tag{2}$$

Here, $y_{r,i}$ corresponds to an element of the vector $Y_r$. It is the system objective function that is computed as a single output item in the $r$th BB.

6) Compute D($\Phi$, $Z$) for use with system optimization, where $\Phi$ is the system optimization objective function. This computation involves the optimum sensitivity derivatives.

7) Solve the system optimization to get $\Delta Z_{\text{opt}}$. The system optimization problem can be stated in the following form:

$$\begin{aligned} &\text{Given:} && Z \text{ and } \Phi_0 \\ &\text{Find:} && \Delta Z \\ &\text{Minimize:} && \Phi = \Phi_0 + \mathrm{D}(y_{r,i}, Z)^T \Delta Z \\ &\text{Satisfy:} && \text{Side constraints on } \Delta Z \end{aligned} \tag{3}$$

where $\Phi_0$ is obtained from the previous SA.

8) Update $X$ and $Z$ and repeat from step 2.

If the starting point is feasible, then the BLISS procedure will maintain feasibility while improving the system objective. Alternatively, if the starting point is infeasible, the constraint violations are reduced while minimizing the increase in system objective, in the problem defined by Eq. (1).

## III. BLISS with Response Surfaces

The use of response surfaces with the BLISS method for system-level optimization will 1) replace the need for subsystems' optimum sensitivity analysis [no D($\Phi$, $Z$) computations], and 2) eliminate the need for subsystem optimizations to yield a feasible solution (and extrapolation issues concerning switching of active subsystem constraints) for each BLISS cycle. In addition, the smoothing operation resulting from the use of response surfaces may improve the convergence characteristics of the numerical optimization scheme, as well as reduce the possibility of being trapped in a local minimum.

In this paper, the response surfaces are used only with the system optimization task and are constructed in the system design variables $Z$ space. They are not used within the subsystem optimizations (BBOPT). Two algorithms, BLISS/RS1 and BLISS/RS2, which are modifications of the original BLISS outlined in the preceding section are proposed. The primary difference between the two algorithms is that in BLISS/RS1 the response surfaces are constructed and updated using system analysis data (step 2 of BLISS procedure, Sec. II), whereas in BLISS/RS2 the response surfaces are constructed using the subsystem (black box/disciplinary) optimization data (step 4 of BLISS procedure, Sec. II) performed for linearly extrapolated $Y$ variables. A flowchart of the BLISS procedure with response surfaces is shown in Fig. 1.

### A. BLISS/RS1: Algorithm 1

The BLISS/RS1 algorithm begins with a complete system analysis to compute the state variables $Y$ and to ensure multidisciplinary compatibility at the start of each BLISS cycle. A local sensitivity analysis is then performed within each subspace/BB to compute the local sensitivities d($Y$, $X$) and d($Y_{r,s}$, $Y_r$). The local sensitivity analysis could be performed using an analytic, semianalytic or finite difference method. The local sensitivities are then used with the system sensitivity analysis to formulate the global sensitivity equations. The solution of the global sensitivity equations provide for the complete derivatives D($Y$, $X$). The state variables computed from the system analysis and the global sensitivities are used with the solution of the $r$ subspace optimization (BBOPT) problems for determining the changes ($\Delta X$) in the local variables. The next stage in the BLISS/RS1 algorithm is the construction and/or the update of the response surfaces for the system-level objective and constraint functions. The response surfaces are constructed in the $Z$ (system variable) space by performing a complete system analysis for each $Z_j$ generated randomly or using an experimental design [design of experiment (DOE)] procedure. Having constructed the response surfaces, these are then used with the solution of the system optimization problem for determining the changes ($\Delta Z$) in the system variables. Finally, the $X$ and $Z$ variables are updated and the BLISS cycle is repeated until a satisfactory convergence is obtained.

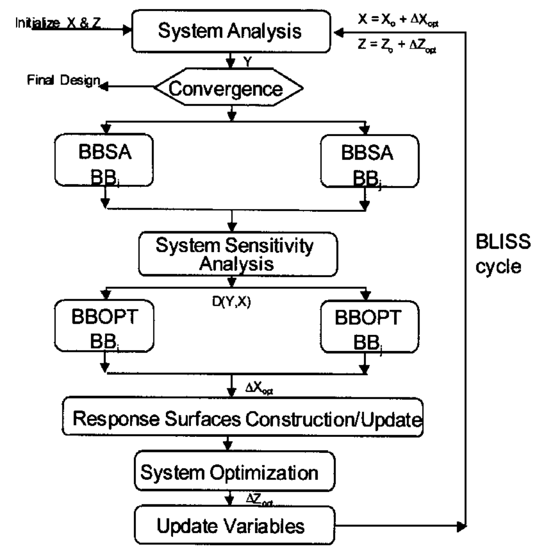


**Fig. 1 Flow diagram of BLISS with response surfaces method.**

A step-by-step definition of the BLISS/RS1 algorithm is provided next.

1) Initialize local disciplinary $X$ and system-level $Z$ variables.

2) Perform SA to compute the state variables $Y$ and the design constraint functions $G$; the SA includes all of the local disciplinary analysis for all of the BBs.

3) Check for convergence of the BLISS procedure.

4) Perform black box sensitivity analysis to obtain d($Y$, $X$) and d($Y_{r,s}$, $Y_r$); perform system sensitivity analysis to compute D($Y$, $X$) only.

5) Maintain constant $Z$, and perform BBOPT in the $X$ space for each BB. This BBOPT is the same as in the original BLISS procedure (outlined in Sec. II).

Given $X$, $Z$ from step 8 and $Y$ from SA in step 2:

$$\begin{aligned} &\text{Find } \Delta X \text{ that} \\ &\text{Minimizes:} && \phi = \mathrm{D}(y_{r,i}, X)\Delta X \\ &\text{Satisfy:} && G(X, Y, Z) \le 0 \end{aligned} \tag{4}$$

Here $y_{r,i}$ corresponds to an element of the vector $Y_r$. It is the system objective function that is computed as a single output item in one of the BBs (or disciplines). The optimal $X$ is used in step 8.

6a) If this is the first pass (first cycle), operate in $\Delta Z$ space, generate $\Delta Z_j$, $j = 1, N$ vectors using DOE methods or randomly, within reasonable move limits on $\Delta Z$.

i) Perform SA for each $\Delta Z_j$, $j = 1, N$, holding $X$ not changed.

ii) Generate response surfaces for the system objective $\Phi$ and each of the system constraints $G$ in the $Z$ space.

6b) If this is the second or subsequent cycle:

i) Perform SA for $X$ and $Z$ updated in step 8.

ii) Optionally, add $\Delta Z_j$, $j = 1, N$ vectors generated randomly or by a DOE method.

iii) If any $\Delta Z_j$ were added in step 6b.ii, repeat SA for these $\Delta Z_j$ while holding $X$ as updated in step 8.

iv) Update the previously generated response surface using the results from SA in step 6b.i and from step 6b.iii.

7) Given the response surfaces for $\Phi$ and special constraints $G_{xz}$ from step 6a.ii or updated in step 6b.iv, perform optimization in the $Z$ space:

$$\begin{aligned}&\text{Find } \Delta Z \text{ that}\\ &\text{Minimizes:} \quad \Phi\\ &\text{Satisfy:} \quad G_{xz} \leq 0, \text{ and } \Delta Z \text{ within move limits}\end{aligned} \tag{5}$$

Note that the special constraints $G_{xz}$ are those constraints that are strongly dependent on both $X$ and $Z$ variables.

8) Update $X$ and $Z$ using the results from step 5 and 7.

### B. BLISS/RS2: Algorithm 2

A step-by-step definition of the BLISS/RS2 algorithm is provided next. As mentioned earlier, the primary difference between this algorithm and BLISS/RS1 is the procedure used for constructing the system objective and constraint response surfaces in the $Z$ space. With BLISS/RS2, the response surfaces are constructed using the subspace optimization results performed for linearly extrapolated $Y$ variables.

1) Initialize local disciplinary $X$ and system-level $Z$ variables.

2) Perform SA to compute the state variables $Y$ and the design constraint functions $G$; the SA includes all of the local disciplinary analysis for all of the BBs.

3) Check for convergence of the BLISS procedure.

4) Perform black box sensitivity analysis to obtain $\mathrm{d}(Y, X)$, $\mathrm{d}(Y_{r,s}, Y_r)$, $\mathrm{d}(G, Z)$, and $\mathrm{d}(G, Y)$; perform system sensitivity analysis to compute $\mathrm{D}(Y, X)$ and $\mathrm{D}(Y, Z)$.

5) Black box optimization and response surface update:

a) If this is the first pass (first cycle), operate in $\Delta Z$ space, generate $\Delta Z_j$, $j = 1, N$ vectors using DOE methods or randomly, within reasonable move limits on $\Delta Z$.

i) For each $\Delta Z_j$, extrapolate $Y = Y$ (from step 2) + $\mathrm{D}(Y, Z)\Delta Z_j$.

ii) Perform BBOPT in the $X$ space for each BB that produces the objective function. This BBOPT is the same as in the original BLISS procedure (outlined in Sec. II), with the exception of $Y$ being tied to $Z$ through step 5a.i.

Given $X$ (from step 2), $Z = Z$ (from step 2) + $\Delta Z_j$ (from step 5a), and $Y = Y(Z)$ (from step 5a.i):

$$\begin{aligned}&\text{Find } \Delta X \text{ that}\\ &\text{Minimizes:} \quad \phi = \mathrm{D}(y_{r,i}, X)\Delta X\\ &\text{Satisfy:} \quad G(X, Y, Z) \leq 0\end{aligned} \tag{6}$$

As stated in the preceding section, $y_{r,i}$ corresponds to an element of the vector $Y_r$ and this corresponds to the system objective function that is computed as a single output item in one of the BBs (or disciplines).

iii) Generate response surfaces for each $\phi$ of each BB in the $Z$ space and response surfaces for the special constraints $G_{xz}$ that are direct functions of $X$ and $Z$.

b) If this is the second or subsequent pass (cycle):

i) Repeat step 5a.i and 5a.ii only for the single $\Delta Z$ obtained in step 6 or optionally for additional $\Delta Z$ vectors generated randomly or by a DOE method.

ii) Update previously generated response surfaces for each $\phi$ and $G_{xz}$ to accommodate new data corresponding to the design points corresponding to the $\Delta Z$ used in step 5b.i.

6) Given the response surfaces for $\phi$ and special constraints $G_{xz}$ for each BB, obtained in step 5a.iii in the first cycle, or from step 5b.ii in subsequent cycles:

$$\begin{aligned}&\text{Find } \Delta Z \text{ that}\\ &\text{Minimizes:} \quad \textstyle\sum_i(\Delta\phi \text{ from BB}_i);\ i = 1 \ldots \text{ all BBs}\\ &\text{Satisfy:} \quad G_{xz} \leq 0, \text{ and } \Delta Z \text{ within move limits}\end{aligned} \tag{7}$$

7) Update $X$ and $Z$ and begin next cycle from step 2.

Needless to mention, the quality of the response surfaces is critical to improving the computational efficiency of the BLISS procedure (or alternatively reducing the number of BLISS cycles). The actual procedure of the response surface construction is outlined in Sec. III.C.

### C. Response Surface Construction

In this work, an "adaptable" response surface model implementation[12] is used for approximation of the optimization responses. In this approach, a minimum number of designs are used to construct an initial approximation model around the baseline design. Typically, a linear approximation model is constructed initially, although the user has an option to request a quadratic initial model. For a linear model, this number would be ($N_{\mathrm{inp}} + 1$), where $N_{\mathrm{inp}}$ is the number of inputs. After the best design is found using this approximation model within the specified design space bounds, the design is analyzed using the "exact analysis," the data are included into the model data set, and the approximation model is regenerated (updated). The exact analysis herein refers to the use of original analysis codes for evaluation of the design point as against using the approximation model to evaluate the design point. The cycle is repeated with new design space bounds and the model is updated with another optimum design for the current model state. Each additional design in the model data set allows for the definition of one additional quadratic term in the polynomial, up to a full quadratic, after which a least-squares fit is used for calculating the coefficients. Because the initial designs constitute only a small fraction of the total data set of the model, their effect is diminished and their distribution in the design space is of much less importance than in the case when all designs for model construction are distributed and analyzed up front. This implementation uses randomly generated or DOE-generated designs for the initial model. The described approach allows the model to be built at run time following the path of the optimizer, and automatically provides more designs for the model near the region of the optimum, resulting in the increased accuracy of the model near the optimum design. In most simple problems convergence occurs before a full quadratic polynomial is constructed or soon thereafter. In more complicated problems with functions of nontrivial shape, restarting of optimization and regenerating of the response surface model may still be required. The algorithm proved to be very efficient and reliable and was tested on several realistic design problems.

The order in which the quadratic coefficients of the model polynomial are defined is determined by the order of input parameters of the model. As more and more design points become available, diagonal quadratic terms are first calculated, and then mixed coefficients are defined. The response surface model performance can be improved by using the results of a DOE study and analysis of variance to determine the most important input parameters, and then use that information for setting the order of defining the model coefficients.[13]

### D. BLISS Enhancements

With the BLISS method, because of the solution of separate optimization problems in $X$ and $Z$ spaces, in the presence of nonconvex constraints, it is possible that a gradient-based optimization search can drive the variables in a different direction as compared to a direction when the design variables are not partitioned. This may result in the BLISS procedure terminating at a different solution point. A possible two design variable scenario is shown in Fig. 2. The local variable is thickness $X$ and the system variable is length $Z$.

Point 1 corresponds to the initial infeasible starting point with respect to both constraints G1 and G2. Point 2 corresponds to the optimal solution point if both variables are changed simultaneously as done in an A-i-O method. Point 3 corresponds to the solution point resulting from a BBOPT (discipline/subsystem) with thickness as the $X$ variable. Finally, Point 4 corresponds to solution point from the system optimization with length as the $Z$ variable. Clearly, BLISS partitioning would end up in a local point as the final solution (when using a gradient-based optimization procedure) as compared to P2.

An approach that is suggested here to handle this situation is similar to the collaborative optimization procedure of Braun and Kroo.[7] Specifically, the relevant $Z$ design variables are duplicated in the BBOPT problems by $X'$ variables and a compatibility constraint is used in the system optimization problem for each duplicated variable to ensure compatibility between $Z$ and $X'$ as the BLISS procedure converges.

## IV. iSIGHT Framework

The BLISS/RS1 and BLISS/RS2 methods are implemented using the MDO language in iSIGHT framework.[14] iSIGHT is a computer-aided optimization framework that provides a customizable and flexible MDO language for integrating simulation tools, analysis programs, and custom optimization methods. It provides for a range of problem-solving capabilities including DOE methods, approximation methods, and optimization search strategies.

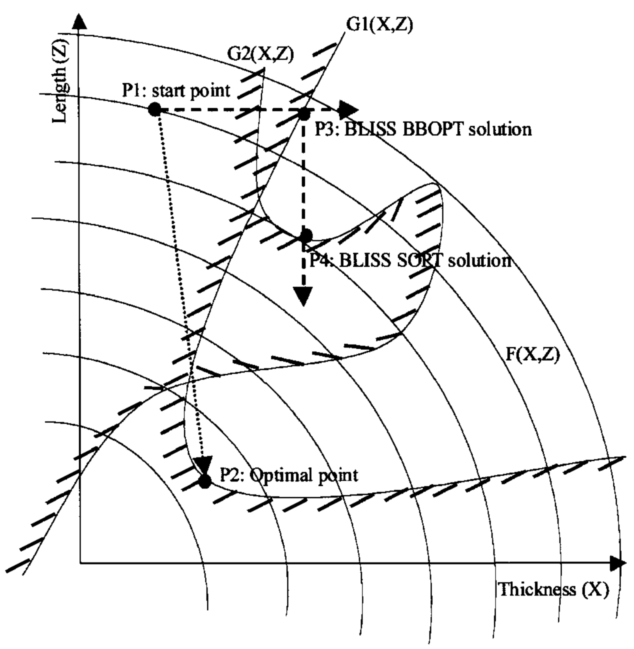


**Fig. 2 Representative design space.**

## V. Numerical Examples

Two design examples are used to test and demonstrate the BLISS procedure with response surfaces. Both the aircraft design optimization[11] and the conceptual ship design problem[15] use low fidelity analysis codes representative of a conceptual design stage. The results from the BLISS/RS1 and BLISS/RS2 methods are compared with the conventional A-i-O and original BLISS methods.

### A. Aircraft Optimization

In this example, a supersonic business jet modeled as a coupled system of structures (BB1), aerodynamics (BB2), propulsion (BB3), and aircraft range (BB4) is used. This problem is identical to the one used in Ref. 11, and complete details of the problem can be obtained from the same reference. A data flow diagram of the coupled system analysis is shown in Fig. 3. The mathematical formulation of the A-i-O optimization problem is as follows.

Maximize: Aircraft range ($F(X)$)

Subject to constraints on:

Stress on wing < 1.09; ($G_j(X)$, $j = 1, 5$)

0.96 < wing twist < 1.04; ($G_j(X)$, $j = 6, 7$)

Pressure gradient < 1.04; ($G_j(X)$, $j = 8$)

0.5 < engine scale factor < 1.5; ($G_j(X)$, $j = 9, 10$)

Engine temperature < 1.02; ($G_j(X)$, $j = 11$)

Throttle setting < $T_{\mathrm{UA}}$; ($G_j(X)$, $j = 12$)

There are a total of 10 design variables $X$, including thickness/chord ratio, altitude, Mach number, aspect ratio, wing sweep, wing surface area, taper ratio, wingbox cross section, skin friction coefficient, and throttle. The A-i-O problem is solved using the sequential quadratic programming (DONLP) implementation in iSIGHT.

The BLISS decomposition consists of four subsystems, including structures, aerodynamics, propulsion, and range. A total of six system design variables are considered: thickness/chord ratio, altitude, Mach number, aspect ratio, wing sweep, and wing surface area. BB1 (structures) has two local variables (X1 = taper ratio,

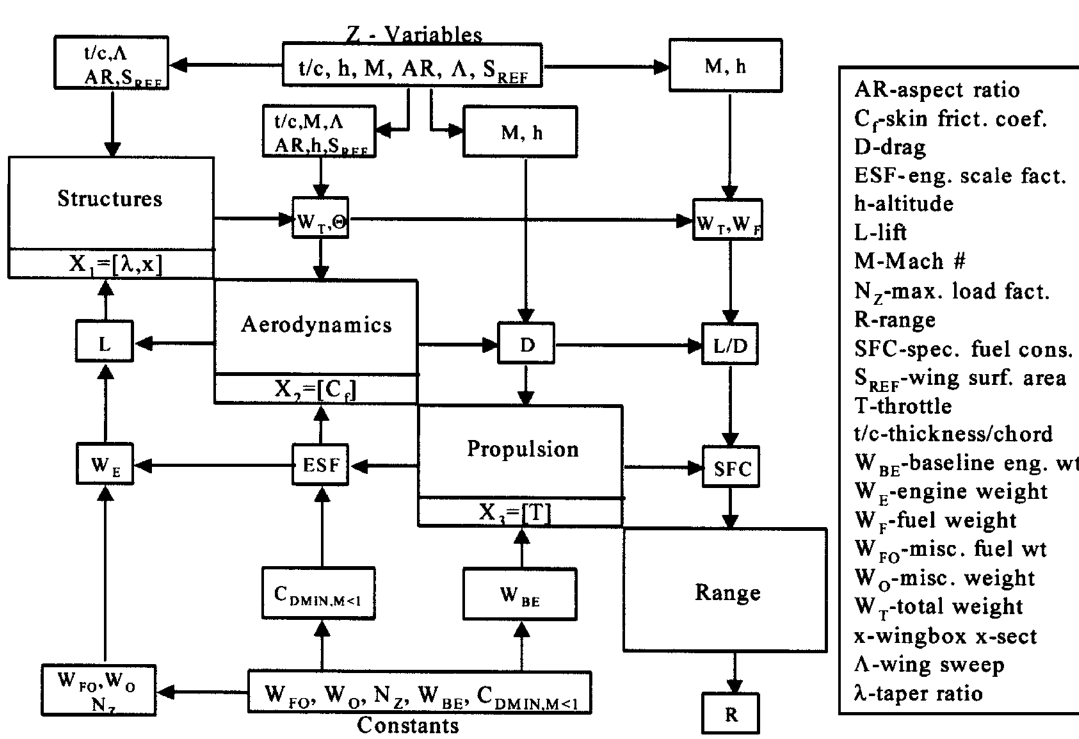


**Fig. 3 Data flow diagram for aircraft MDO problem.**

**Table 1 Aircraft optimization results**

| Case | Initial objective | Initial maximum constraint | Final objective | Final maximum constraint | Computational effort: Number of system analyses | Computational effort: Number of subsystem analyses |
|---|---|---|---|---|---|---|
| A-i-O | 535.79 | −0.162 | 3964.19 | +1.0*e*−08 | 119 | 1428 |
| A-i-O/RS | 535.79 | −0.162 | 3974.84 | +0.0013 | 72 | 864 |
| BLISS | 535.79 | −0.162 | 3964.07 | +1.92*e*−05 | 7 | 491 |
| | | | | | (6 BLISS cycles) | |
| BLISS/RS1 | 535.79 | −0.162 | 3961.50 | +0.0 | 17 | 354 |
| | | | | | (4 BLISS cycles) | |
| BLISS/RS2 | 535.79 | −0.162 | 3964.12 | +0.0 | 12 | 1097 |
| | | | | | (11 BLISS cycles) | |

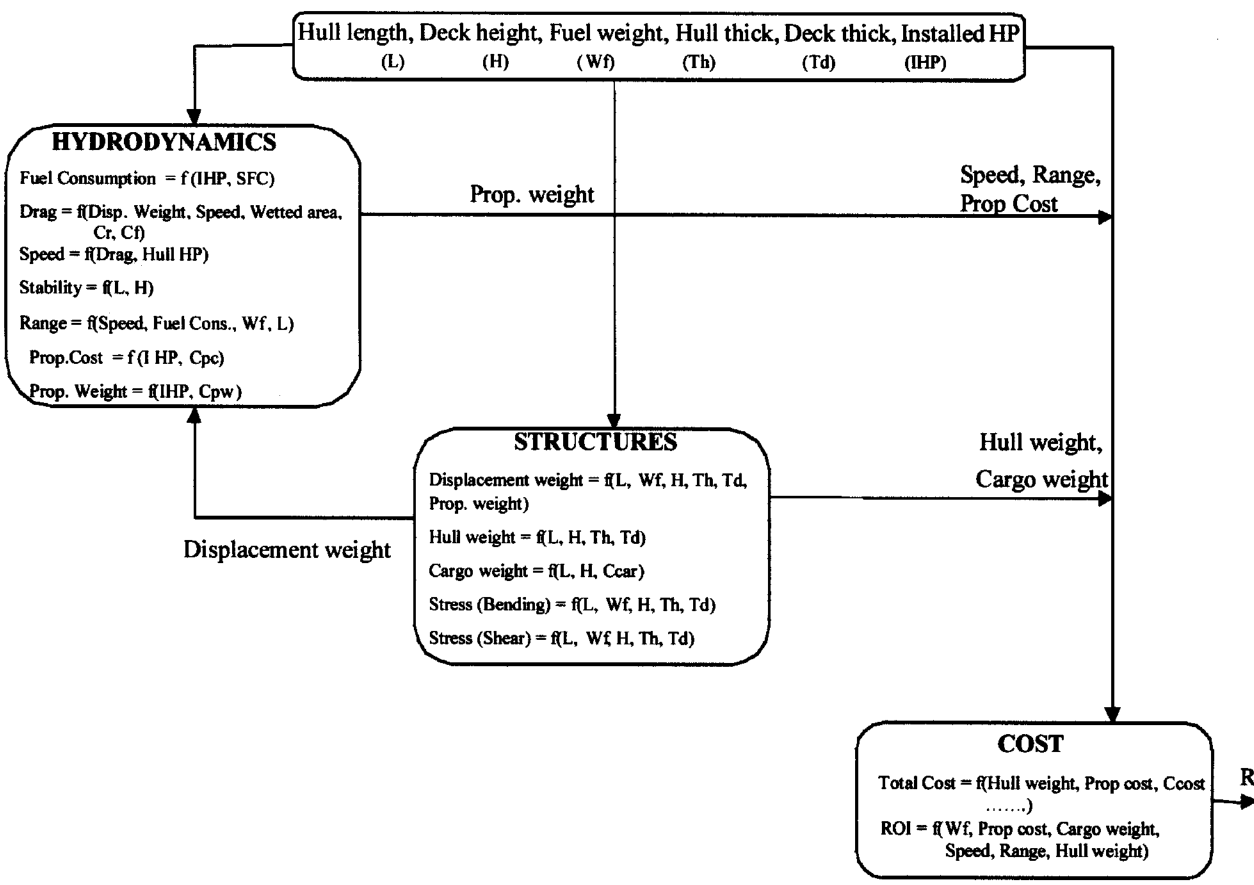


**Fig. 4 Data flow diagram for conceptual ship design problem.**

wingbox cross section), BB2 (aerodynamics) has one local variable (X2 = skin friction coefficient), and BB3 (propulsion) has one local design variable (X3 = throttle). BB4 computes the system objective range and does not perform any local optimization.

The results obtained from BLISS/RS1 and BLISS/RS2 methods are compared with A-i-O, A-i-O/RS and original BLISS methods in Table 1. The A-i-O/RS is a sequential approximation-based all-in-one optimization strategy that involves the use of response surface model for approximate evaluations of the design objective and constraint functions.

### B. Conceptual Ship Design

MDO of a conceptual design of an oil tanker ship, where several disciplines are analyzed to provide one complete system analysis is considered. This problem is identical to the one used in Ref. 15. The disciplines involved in the system analysis include the following.

1) Hydrodynamics (BB1): involves engine propulsion calculations, wave and skin resistances (drag) modules, stability factor and range calculations;

2) Structures (BB2): involves weights and stress calculations.

3) Cost (BB3): total ship cost and the return-on-investment (ROI) computations.

The mathematical formulation of the A-i-O optimization problem is as follows.

Maximize: ROI $= f(X)$, subject to range = 10,000 Nm ($\pm 1.0\%$), displacement weight $= 2 \times 10^8$ lb ($\pm 1.0\%$), maximum (bending and shear) stress < 30 ksi, stability factor < 0.0, and bounds on design variables. Six design variables, including hull length, deck height, hull thickness, deck thickness, installed engine horse power, and fuel weight are considered.

The BLISS decomposition consists of 3 subsystems (hydrodynamics, structures, and cost). The system-level objective is to maximize ROI. A total of three system design variables are considered: hull length, deck height, and fuel weight. BB1 has one local variable (X1 = installed HP) and local constraints on range. BB2 has two local variables (X2 = hull thickness and deck thickness) and local constraints on displacement weight, bending, and shear stresses. BB3 computes the ROI and does not perform any local optimization. All of the local constraints and stability requirement computed in BB1 are treated as system-level constraints.

Figure 4 shows a data flow diagram of one full system analysis for the problem. The results are provided in Table 2. The initial design is an infeasible design with an ROI of 0.2660. The ROI is a scalar quantity that represents (1/number of years) to recover the

**Table 2 Conceptual ship design results**

| Case | Initial objective | Initial maximum constraint | Final objective | Final maximum constraint | Computational effort: Number of system analyses | Computational effort: Number of subsystem analyses |
|---|---|---|---|---|---|---|
| A-i-O | 0.2660 | +1.807 | 0.278 | +0.003 | 111 | 333 |
| A-i-O/RS | 0.2660 | +1.807 | 0.278 | +0.002 | 50 | 150 |
| BLISS | 0.2660 | +1.807 | 0.262 | +0.002 | 46 | 5676 |
| | | | | | (45 BLISS cycles) | |
| BLISS/RS1 | 0.2660 | +1.807 | 0.266 | +0.003 | 18 | 367 |
| | | | | | (14 BLISS cycles) | |
| BLISS/RS2 | 0.2660 | +1.807 | 0.270 | +0.003 | 10 | 756 |
| | | | | | (9 BLISS cycles) | |

investment. In this example, the BLISS method did not arrive at the best known solution of 0.278 for the objective function (ROI).

## VI. Summary

A method for optimization of engineering systems has been developed and demonstrated on numerical examples of an aircraft and a ship. Typically, the design variables cluster into a set of a relatively few design variables that govern the system design, and a set of local variables that govern the design detail. The latter are usually large in number, but they cluster by the system components. The method decomposes the problem into several optimizations at the component level that may be executed concurrently and a coordinating optimization at the system level. In the original version of the method,[9] the system-level optimizations were linked to the component-level optimizations by the optimum sensitivity derivatives. In the version reported herein the two optimization levels link through the response surfaces of a polynomial function type, and two variants of that linkage are introduced. In variant 1, the response surfaces for the system objective and the system constraints are constructed in the space of the system design variables using the system analysis results. In variant 2, the response surfaces are being updated with the system component optimization results. In yet another variant of the method, the system-level design variables are assigned counterparts at the component level, with special constraints dedicated to enforce compatibility of each variable and its counterpart. The purpose of the preceding variations of the method algorithm is to improve the method robustness by smoothing discontinuities and overcoming the detrimental effects of nonconvexity.

The method was demonstrated in application to a conceptual-level design of a supersonic business jet aircraft and a ship. Results were compared to those obtained by an all-in-one optimization (A-i-O), A-i-O with the response surfaces, and the original BLISS. In the aircraft test case, the method minimum objective agreed very well with that of the benchmark A-i-O. In the ship test case, the method fell short of the benchmark value by 4.3%. In all of the tests, the method showed a satisfactory capability to satisfy the constraints. In regard to the amount of numerical work, two different metrics were used. The metric equated to the number of the system analyses was found to be case dependent. By that metric in the aircraft application, the method was not as efficient as BLISS but still an order of magnitude more efficient than A-i-O. In the ship case, the method was both an order of magnitude more efficient than A-i-O and about twice more efficient than the original BLISS. The other metric was the number of individual component analyses. Under that metric in the aircraft case, the method was more expensive than the original BLISS but still more economical than A-i-O, whereas in the ship application the method turned out to be more efficient than BLISS and about on par with A-i-O.

When interpreting the preceding results, one should observe that the aircraft case is a strongly coupled system. Figure 3 shows the number of instances where data are being passed between the modules in both directions, forward and back. In contrast to that, the ship system, as illustrated in Fig. 4, is relatively weakly coupled with only one instance of the data feedback (displacement weight). In fact, if that particular data feedback were absent, the system would be purely hierarchical. In the coupled system, the original formulation of BLISS has some advantage because the optimum sensitivity data enable the system-level optimization to anticipate better the effect of the system variable changes on the modules. In a hierarchical system, in general, a two-level optimization converges faster hence the absence of the optimum sensitivity data as a link between the levels in BLISS/RS appears not to be detrimental.

When assessing computational efficiency of the method, one should also remember that the underlying analyses were exceedingly simple, typical of the conceptual design stage. One expects that the cost of the system analysis relative to the component analysis will increase as the design moved to the preliminary and detailed stages, hence the metric based on the number of the system analysis is likely to dominate.

Finally, the BLISS with response surfaces algorithm is well suited for exploiting the concurrent processing capabilities in a multiprocessor machine. Several steps, including the local sensitivity analysis, local optimization, response surfaces construction, and updates, are all ideally suited for concurrent processing. Needless to mention, such algorithms that can effectively exploit the concurrent processing capabilities of the compute servers and computer technology now becoming available will be a key requirement for solving large-scale industrial design problems.

## Acknowledgments

The first author's work was supported by NASA Contract L68259D. The support of Charles Yuan of Engineous Software, Inc., with setting up the iSIGHT problem description and translation of NASA aircraft design codes from MATLAB scripting language to C is acknowledged.

## References

[1] Sobieszczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems," *Proceedings of the 2nd NASA/USAF Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, NASA CP-3031, 1988; also NASA TM-101494, 1988.

[2] Balling, R. J., and Sobieszczanski-Sobieski, J., "Optimization of Coupled Systems: A Critical Overview of Approaches," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 6–17.

[3] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization*, Vol. 14, No. 1, Aug. 1997, pp. 1–23.

[4] Cramer, E. J., Dennis, J. E., Frank, P. D., Lewis, R. M., and Shubin, G. R., "Problem Formulation for Multidisciplinary Design Optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.

[5] Renaud, J. E., and Gabriele, G. A., "Improved Coordination in Non-Hierarchic System Optimization," *AIAA Journal*, Vol. 31, No. 12, 1993, pp. 2367–2373.

[6] Renaud, J. E., and Gabriele, G. A., "Approximation in Non-Hierarchic System Optimization," *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 198–205.

[7] Braun, R. D., and Kroo, I., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," *Multidisciplinary Design Optimization, State of the Art*, edited by N. Alexandrov and M. Y. Hussaini, Society of Industrial and Applied Mathematics, Philadelphia, 1997.

[8] Sobieski, I. P., and Kroo, I., "Collaborative Optimization Using Response Surface Estimation," AIAA Paper 98-0915, Jan. 1998.

[9] Sobieszczanski-Sobieski, J., Agte, J., and Sandusky, R., Jr., "Bi-Level Integrated System Synthesis (BLISS)," *Proceedings of the AIAA/USAF/NASA/ISSMO 7th Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, 1998, pp. 1543–1557.

[10]Box, G. E. P., and Draper, N. R., *Empirical Model Building and Response Surfaces*, Wiley Series in Probability and Mathematical Statistics, Wiley, New York, 1987.

[11]Sobieszczanski-Sobieski, J., Agte, J., and Sandusky, R., Jr., "Bi-Level Integrated System Synthesis (BLISS)," NASA TM-1998-208715, Aug. 1998.

[12]Golovidov, O., Kodiyalam, S., Marineau, P., Wang, L., and Rohl, P., "A Flexible Object-Based Implementation of Approximation Models in an MDO Framework," *Design Optimization: International Journal for Product and Process Improvement*, Vol. 1, No. 4, 1999, pp. 388–404.

[13]Kodiyalam, S., Su, J., and Wujek, B., "Design of Experiments Based Response Surface Models for Design Optimization," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 39th Structures, Structural Dynamics, and Materials Conference*, AIAA, Reston, VA, 1998 (AIAA Paper 98-2030).

[14]*iSIGHT Designer's and Developer's Guide*, *Version 4.0*, Engineous Software Inc., Morrisville, NC, 1998.

[15]Kodiyalam, S., Swenson, L., and Stehlin, B., "Multidisciplinary Design Optimization with Object-Oriented Product Modeling," *Proceedings of the Optimization in Industry Conference—I*, Engineering Foundation, New York, 1997, pp. 53–56.

A. Messac
*Associate Editor*