JOURNAL OF AIRCRAFT
Vol. 36, No. 1, January–February 1999

# Framework for Multidisciplinary Design Based on Response-Surface Approximations

Stephen M. Batill* and Marc A. Stelmack†
*University of Notre Dame, Notre Dame, Indiana 46556*
and
Richard S. Sellar‡
*The Summit Group, Inc., Mishawaka, Indiana 46545*

**A framework has been proposed to allow for the multidisciplinary design of coupled, nonhierarchic systems. This approach is based on the ability to decompose a model-based analysis of a coupled system into subspaces or contributing disciplines. These subspaces are defined in terms of the design variables that they can influence and the information they contribute to the characterization of the complete system. The subspace coupling is based on the information they exchange. By using a set of response surface approximations, experts responsible for a particular subspace can make design decisions with the goals of improving the complete system merit and satisfying system constraints. Because design variables can be shared between subspaces, coordination of the subspace design decisions is achieved by the solution of a fully approximate optimization problem involving the complete set of system design variables. The implementation of this framework using two flight vehicle concept design problems is presented.**

## Introduction

THE foundation for multidisciplinary design optimization (MDO) was established with parallel developments in computing, model-based engineering analysis, particularly structural analysis, and optimization methodology that began to coalesce in the 1960s. The influence of these developments has since expanded from early applications in aerospace vehicles to include many other complex engineering systems. Current research continues to focus on improvements in automated analysis (developing information to assist in decision making) and optimization methods (decision-making processes), and integrating these activities into the engineering design process using appropriate computing technologies. The purpose of this paper is to describe a basic framework in which model-based design decisions can be made for complex systems governed by information-coupled technical disciplines. Aerospace vehicles still represent some of the most challenging of multidisciplinary systems, and they provided the motivation for this research. They are used in the applications presented here.

At the onset, the authors wish to emphasize an attempt on their part to differentiate between engineering system design and mathematical optimization. Though it would be the goal of any designer to develop the global optimum design, in most practical situations this is generally not possible. Therefore, one is often satisfied with a design that meets the primary constraints or performance goals and is more effective by some quantifiable measure, i.e., merit function, than other designs that have been considered. It is also important that this design be identified in a reasonable amount of time using available resources. The goal of the framework presented in this paper is to present a methodology for the efficient selection of a feasible design for a multidisciplinary system and to demonstrate issues of problem formulation and implementation using two simple applications.

### Terminology

To present the discussion in this paper, a number of terms or concepts need to be developed. These terms and concepts are used throughout the paper.

Design variables, $\{x\}$: The set of independent parameters that can be controlled by the designer. The complete set of design variables defines the design at its current level of abstraction.

State variables, $\{y\}$: A set of parameters that are used to describe the performance or characteristics of the current design. Design decisions are based on state information. States are functions of design variables and are often determined using computer-based models. The computer models may be as simple as numerical evaluation of explicit analytic expressions, or more complex numerical models such as those based upon computational fluid dynamics (CFD) or finite element method (FEM) analyses. The distinction between design and state variables is problem dependent. In a mission definition study, aircraft range could be a design variable, where in other design studies, range would normally be considered a state.

System analysis, SA: That process whereby the complete set of state variables associated with a specific set of design variables is determined by satisfying a set of state equations

$$\{y\} = F\{x\} \tag{1}$$

For multidisciplinary applications, the system analysis involves a variety of different engineering disciplines, e.g., aerodynamics, structures, performance, stability and control, etc., and their associated analyses methods and corresponding models. The exchange of information between the models and analysis tools and solution of this set of typically coupled, nonlinear equations are central issues in MDO.

Consistent design: A set of design variables and associated states that satisfies all of the conditions set forth in the state

Received Dec. 11, 1997; revision received March 5, 1998; accepted for publication Aug. 9, 1998. 

*Professor, Aerospace and Mechanical Engineering. Associate Fellow AIAA.

†Graduate Research Assistant, Aerospace and Mechanical Engineering. Student Member AIAA.

‡Consultant. Senior Member AIAA.

equations in the system analysis. Not every set of design variables may result in a unique, consistent design.

Feasible design: A consistent design that satisfies a set of explicit requirements or constraints in addition to the state equations that are imposed on either design variables or states. In most optimization problems the constraints are explicit functions of design variables and system states.

Optimum design: A feasible design with the best performance as quantified by some measure of merit relative to all other designs proximate to the optimum. This measure of merit is usually some combination of system states and design variables. The optimum can either be local or global, and for most practical problems it occurs on numerous constraint boundaries.

As the MDO discipline matures, a more widely accepted notation will evolve, but until that time, individual groups and organizations will establish their own working vocabulary. One particularly useful effort to help establish a classification and associated terminology for coupled systems was developed by Balling and Sobieszczanski-Sobieski[1]; reference is made to a number of their concepts in the current paper.

### System Analysis

The SA is the source for the information used to predict system performance and make design decisions. In multidisciplinary systems, the formulation of the system analysis can be extremely complex. As an example, traditional preliminary aircraft design methodologies often provided recipes for aircraft concept sizing. Raymer[2] provides such a methodology and examples for aircraft concept sizing. The sequence of events in this process is selected to minimize interdisciplinary couplings, is iterative in nature, and is not guaranteed to result in feasible designs. As attempts are made to include more complex analysis models into the process, this type of approach cannot be used (nor was it ever intended for that purpose). With increasing design detail and the desire to include the influence of more complex, model-based discipline-specific analyses, the process of conducting the system analysis becomes much more difficult, computing resource intensive and time consuming.

Early MDO efforts and significant recent work have focused on automating system analyses. Establishing efficient communication between different disciplines' analysis modules (such as aerodynamic loads and finite element structural analysis) is, and will continue to be, a challenge. One useful means of representing the SA is the $N$-square matrix shown in Fig. 1. This is used to illustrate the information dependency between the different subspaces, disciplines, or contributing analyses. The three terms: subspaces, disciplines, or contributing analyses can be used to represent the elements into which the system analysis is decomposed. In this work, the term subspace was selected to represent the most general case, which could bridge discipline or analysis boundaries, and to highlight the decomposition of the complete system into processes that are responsible for developing system-state information. Often this decomposition is along traditional engineering disciplines, but it may also represent groups of disciplines or may be based upon optimal information coupling such as that developed by DeMAID.[3] In the current paper, the assumption has been made that the system has been modeled, the decomposition of the system analysis into subspaces has taken place, and the information couplings have been identified.

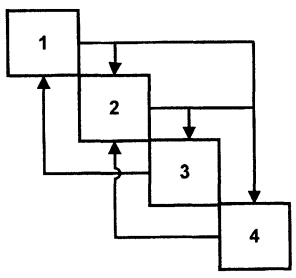


**Fig. 1 $N^2$ dependency matrix.**

Some comments with respect to the coupled system analysis process are required. The purpose of the system analysis is to identify the set of consistent states for a given set of design variables. The couplings that occur can be either natural, such as those associated with static aeroelasticity, i.e., aerodynamic loads depend upon deformation and vice versa, or imposed as the result of including constraints as part of the system equations. If one of the state equations is a range requirement for an aircraft and cruise segment fuel is a design variable, then couplings in the system analysis may be developed, as total weight would depend upon fuel weight, which would depend upon total weight. This illustrates the potential for confusing the roles of the system analysis and optimization. In the past, it was the role of the designer or design team to perform iterations to ensure feasibility of the design, and along the way attempts were made to improve the merit of the design. MDO provides an opportunity to integrate these processes in a single framework, and it is that type of framework that is the focus of this paper.

### Background

To provide a context for the current paper, a brief overview of selected developments related to MDO frameworks for coupled systems is included. A more comprehensive overview can be found in Refs. 4 and 5. The efficient integration of the automated decision-making process, i.e., optimization, and the system analysis is the eventual goal in the development of an MDO framework. The purpose is to provide a rational process for selecting a design from the usually overwhelming number of potential candidates.

A straightforward approach would be to simply wrap the coupled system analysis with an optimization algorithm in a manner in which a complete system analysis is performed any time the optimization algorithm requires state information. This approach in its basic form, referred to as a single-level nested analysis and design (NAND),[1] proves unsuitable except for select cases because of the rapid growth in problem size and cost of repetitive system analyses. One advantage of this approach is that the system analysis should provide consistent designs throughout the process. Thus, once the design proves feasible (often the first step in an optimization algorithm), the process can be terminated when acceptable merit improvement has been realized, but before a true optimum is achieved. This goal is quite consistent with many practical design studies.

An extension of the NAND approach that has proven quite useful involves the use of sequential optimization of approximations to the complete system. This usually involves the use of information developed in a sensitivity analysis, another area of important current interest in MDO research, which is either part of or augmented to the system analysis. The sensitivity information is used to develop low-order approximations to the system behavior in the vicinity of the current design. System optimization is performed using this local approximation and all or a subset of the design variables or constraints. In this case, all design decisions are made at the system level and are based solely on the logic associated with the optimization algorithm. This approach has served to extend the utility of NAND and allow application to problems of significant scope. The automated structural optimization system (ASTROS),[6] developed by the U.S. Air Force, is an example of this approach.

In an alternative single-level approach, referred to as simultaneous analysis and design (SAND), the system analysis and the optimization processes are performed simultaneously. This usually involves the introduction of new design variables and consistency constraints, and formulating an optimization problem with the goal of obtaining a consistent, feasible, and optimal design from a single, iterative process. This approach helps address the problems associated with couplings within the system analysis, but it too is limited by the problem size

and the issue of providing inconsistent and infeasible designs until convergence to an optimum, which is not guaranteed, is achieved.

The two approaches discussed in the previous text involve a single optimization algorithm that operates on the entire system at each step in the process. More recent MDO developments attempt to exploit the decomposition of the system analysis as it is integrated into the optimization process. This often involves the formulation of multiple-level optimization problems. System-level optimization is used for problem coordination or target setting, whereas subspace level optimization is associated with the elements into which the complete system has been decomposed. The optimization problems at each level have their own design variables, constraints, and merit. Two of these are concurrent subspace optimization (CSSO), introduced by Sobieszczanski-Sobieski,[7] and collaborative optimization, proposed by Kroo et al.[8]

The CSSO approach is based on the ability to integrate subspace sensitivity analyses into the system analysis. The sensitivity information is used in conjunction with global sensitivity equations (GSEs),[9] to allow individual subspaces to attempt to optimize the system while performing approximate corrections to coupled states during subspace optimization. Once individual subspaces are allowed to alter the design (modify some subset or all of the design variables) their efforts must be coordinated. The original coordination procedure suggested by Sobieszczanski-Sobieski[7] was based upon responsibility coefficients and alternative formulations have been suggested. Most notable are those of Renaud and Gabriel[10] and Wujek and Renaud,[11] in which the system-level coordination is based on an approximate optimization strategy. In each of the CSSO formulations, individual disciplines are given the opportunity to attempt to improve the design subject to system-level constraints and merit. Coordination of the independent decisions associated with each discipline is established as a separate step. An attractive practical aspect of the CSSO approach is that within each design cycle a complete system analysis is performed, so that one is assured of a consistent design, and premature termination of the optimization process could still provide an acceptable design.

The collaborative optimization (CO) framework of Kroo et al.[8] has certain similarities to SAND approaches, but it attempts to exploit the decomposition of the system. A system-level constrained optimization is performed to provide target values for shared and auxiliary design variables that are used in the subspace level optimizations. A most attractive aspect of this approach is that certain issues associated with the complex coupling in the system analysis are eliminated because the process is intended to assure consistency while achieving an optimum design, though this requires the introduction of auxiliary design variables at the system level. An added drawback is that a consistent design is only guaranteed when the optimization has converged. Thus, the advantages gained by elimination of the requirement for convergence of numerous system analyses must be weighed against potential drawbacks for a given application.

The framework presented in the current paper is an extension of concepts associated with both the NAND and CSSO approaches. It is also based upon the earlier work of Swift and Batill,[12,13] and Batill and Swift,[14,15] and Hajela and Berke[16] using artificial neural networks as response surface approximations. In Ref. 15 and related work, a multidisciplinary design framework was demonstrated in which design variables were split into two groups, system level and discipline specific. Optimum designs at the discipline level, using system-level merit and discipline constraints, were determined using an initial, small set of potential designs described by fixed system-level variables. To select optimum values of the system-level design variables, a second, system-level optimization was performed using an approximation to the discipline design space characterized by optimum designs at the discipline level. An artificial neural network was used to approximate the response surface described by the optimum discipline level designs. This approach allowed for the use of discrete design variables at the system level, another important consideration in the practical implementation of MDO frameworks. Though this approach was based upon the ability to decompose the system into two levels, it lacked generality for truly multidisciplinary applications.

## Response Surface Approximation Based MDO Framework

The approach taken in this paper is based on the fundamental structure of CSSO and on the use of neural network-based response surface approximations. The CSSO approach does not allow for the inclusion of information other than the current design point and local sensitivities to help maintain consistent states during subspace optimization. Other sources of such information could include the characteristics of other designs that have been considered as part of earlier studies. It is desirable to take advantage of this accumulated knowledge in the design process. By restricting the system approximations to simple functions based on local state information and its sensitivity, the inclusion of such prior knowledge is precluded. The use of GSEs restricts the optimization algorithm to be gradient-based and the system design vector to a set of continuous variables only.

For the framework presented herein individual subspace experts can each contribute to the design process by suggesting candidate designs. These designs can be developed using whatever methods or tools are best suited for the particular subspace and can be based on the system-level merit and constraints. This allows the discipline experts the opportunity to explore the design space as they see fit, and suggest new candidate designs to populate the design database. It also attempts to exploit the response surface approximation approaches to provide a means for consistency during subspace design and system-level coordination. To discriminate between previous frameworks, the current approach is referred to as concurrent subspace design (CSD), which describes its evolution and basic purpose. The CSD formulation as implemented in this paper uses three-layer, feedforward, sigmoid-activation neural networks for subspace-level response surface approximations and system coordination.

A schematic of the CSD algorithm is shown in Fig. 2. The process flow in the CSD algorithm begins with the selection of a group of baseline designs. A system analysis is performed on each of these designs to initially populate the design data-

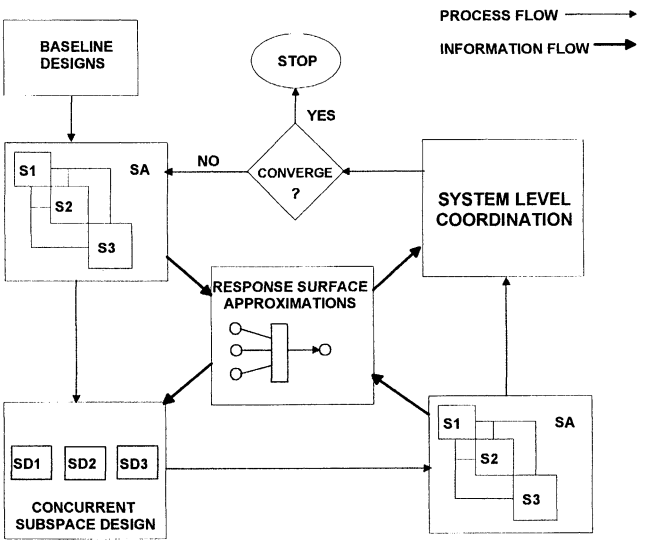


Fig. 2 Schematic of CSD framework.

base with design variable and associated state information. A set of response surface approximations is then formed using this database of designs. These approximations are used to provide nonlocal state information to the subspace experts for their use while they are performing concurrent design/optimization studies. The independent and concurrent design activities of the subspace experts yield additional candidate designs for the database. Subsequently, updated system approximations are developed using this extended database. The response surface approximations are then used, in a manner similar to the approximate NAND approach, to provide system coordination through a single, approximate system-level optimization. The ability of this approach to provide improved or optimum designs is directly related to the accuracy of the response surface approximation used for system-level coordination.

The CSD algorithm is initiated from a design database that can contain a baseline design, perturbations about the baseline, or information from other design studies or data sources. This algorithm allows discipline designers to be creative in arriving at new candidate designs. Creativity implies that the solutions obtained by discipline-level designers can be very different from the current design being considered (not restricted by move limits), and they can suggest as many designs as time and resources will permit. It is desirable to allow designers to propose unique concepts, particularly in the preliminary design stage. The CSD algorithm does not stipulate how the subspace designers select new designs to add to the design database. Designers at the subspace level are not required to optimize at all, but rather provide new designs by any available means.

To effectively participate in the design process, subspace designers in the CSD algorithm are given the ability to change a subset of the system design variables. The complete set of design variables need not be partitioned in a manner in which each design variable is assigned to only one subspace. Designers have available to them information that can be used to approximate the impact of their decisions on nonlocal states at little computational expense once the approximations have been parameterized. Because of the use of the response surface approximations, nonlocal constraints are evaluated in each subspace based on approximate state information. It should be noted that not all subspaces need to develop new design candidates at each step in the process. Some subspaces may not contribute new designs, nor do they need to use the full fidelity analysis tools during this process. More efficient analysis methods or simpler models could be considered during the subspace optimization.

The result of the CSD phase of the CSD algorithm is a set of new designs. These new designs are analyzed using the complete system analysis and added to the design database from which an updated system approximation is constructed. The design database continues to expand with each iteration of the algorithm; each time a new approximation is formed using all of the appropriate design database information. The CSD algorithm relies solely on data that results from system analyses of designs and forms global approximations using all appropriate design data. The primary cost of this approach is the computational and personnel time and effort required to perform a system analysis on each of the resulting designs from the disciplinary design phase.

The next step in the process is the fully approximate system coordination problem. It has the benefit of using all of the system design variables and of removing certain restrictions on the optimization algorithm. Because the cost of performing the approximate system analysis is quite small, optimization algorithms such as those associated with discrete system optimization can be used. An additional benefit of the global approximations is apparent when the design requirements or constraints change during the design process. Because the response surface approximations are reformed using all available design space information, and they depend on states and design variables, the rapid re-evaluation of designs based on a change in design criteria is possible. It is even possible to initiate the system coordination problem from multiple starting locations so that one can identify and avoid local optimum.

The CSD framework is not easily classified using the Balling and Sobieszczanski-Sobieski[1] notation because they did not directly include the concept of design space approximations in the form of global response surfaces. Because it is a multilevel approach allowing for concurrent design and each discipline is provided information that allows it to maintain approximate consistency, it might be classified as a multi-NAND–NAND approach, similar in basic structure to CSSO. It is also obvious that it is similar to the more traditional design methods based upon carpet plots.[2] In CSD, the multidimensional carpet plots have been parameterized in the form of neural network approximations and the most appropriate optimization strategies are used to seek the best design.

## Application and Implementation of CSD

Two MDO applications are used to demonstrate the CSD framework. Each contains certain unique characteristics and each is used to describe in more detail the CSD framework. The first is an aircraft concept sizing (ACS) problem, and the second a vehicle sizing for an autonomous hover-craft (AHC). The purpose of these two studies is to explore issues in problem formulation and CSD framework implementation. These examples do not involve computationally expensive system analyses, and are thus intended to explore issues in CSD problem formulation and not detail improvements in computational costs.

### Aircraft Concept Sizing

An aircraft sizing problem is considered that involves making a preliminary gross weight estimate to achieve mission range requirements and sizing the wing to meet low-speed flight requirements. The analysis tools used are based on empirical models, and the system analysis is shown in Fig. 3, which also illustrates the information exchanged between the subspaces. The analysis problem was decomposed into three subspaces. The system design variables and states associated with each subspace are given next. Design variables: Aspect Ratio (AR), wing area ($S_{wing}$), fuselage length ($L_f$), mean fuselage diameter ($D_f$), cruise altitude ($h_{cruise}$), cruise velocity ($V_{cruise}$), and fuel weight ($W_{fuel}$). System states: $(L/D)_{max}$ (aerodynamics), empty weight ($W_{empty}$) (weights), total weight ($W_{total}$) (weights), range ($R$) (performance), and stall velocity ($V_{stall}$) (performance). There are seven design variables and five system states. As formulated, this system analysis has been adapted for use as part of an optimization process. Detailed development of this demonstration problem is presented in Ref. 17.

Performing a system analysis for a given set of design variables provides a consistent design, but depending on the performance requirements for this aircraft, the design may not be feasible. The two primary performance characteristics, range

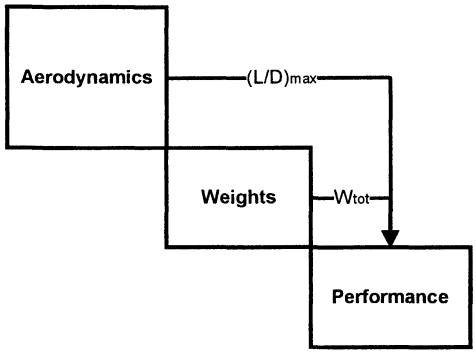


**Fig. 3 Aircraft concept sizing application.**

**Table 1 ACS subspace design variables**

| | Aerodynamics | Weights | Performance |
|---|---|---|---|
| Subspace design variables | AR, $S_{wing}$, $L_f$, $D_f$ | AR, $S_{wing}$, $L_f$, $D_f$, $h_{cruise}$, $V_{cruise}$, $W_{fuel}$ | $S_{wing}$, $W_{fuel}$ |
| Required nonlocal states | —— | —— | $(L/D)_{max}$, $W_{total}$ |
| Computed states | $(L/D)_{max}$ | $W_{empty}$, $W_{total}$ | $V_{stall}$, range |

and stall speed, are computed as states in the system analysis, and unless required values of these states are included as state equations, there is no guarantee that a given design would meet these requirements. If these requirements were included as state equations, a more complex system analysis with feedback coupling would result and the system analysis would require an iterative process. Because these requirements are imposed as part of the design optimization process, feedback is eliminated and the system analysis does not require iterative solution. This is an example of a system in which feedback would result from a modification of the problem statement, not natural coupling, as demonstrated in the next application. In this example, feedback is avoided by the form of the problem statement.

The optimization problem for the MDO formulation of the ACS problem would take the following form. Determine the set of design variables ($\boldsymbol{x}$) which will

Minimize:

$$W_{total} = W_{empty} + W_{fuel} + W_{payload} \tag{2}$$

Subject to:

$$\frac{\text{Range}}{\text{Range}_{required}} - 1 \geq 0.0$$

$$1 - \frac{V_{stall}}{V_{stall\ required}} \geq 0.0$$

$$x_{lower_i} \leq x_i \leq x_{upper_i}$$

where the empty weight, range, and stall speed are functions of the design variables. Qualitative review of the resulting design space indicates that, in general, total weight varies linearly with each design variable. The range and stall constraints are nonlinear functions of fuselage size and fuel weight. The constraints vary linearly with the other design variables.

Formulation of subspace optimization/design problems allows for the concurrent design of the system by allowing designers to solve approximate system optimization problems at the subspace level. The design variable allocation for each of the subspace design (optimization) problems and subspace or local states are given in Table 1. Each subspace optimization problem takes the same form as Eq. (2), except approximate values for the nonlocal states are used as required.

Note that the aerodynamics subspace has as design variables the geometric properties of the aircraft, and generates a state, $L/D_{max}$, which does not explicitly appear in either the merit function or constraints. Consequently, in this CSD formulation, the design problem solved in the aerodynamics subspace must be completely approximate. The inclusion of aerodynamics in the subspace design phase is problematic because the contribution of aerodynamics to the design process is to provide state information to other disciplines. This is true of many disciplines in the engineering design process. The weights subspace design problem requires the approximation of the system constraints and allows for direct computation of the merit function. The empty weight is based solely upon design variables and fixed parameters, i.e., ultimate load factor, payload weight, etc. The design problem solved in the performance subspace involves the approximation of the merit function and direct determination of the constraints.

In this simple application, three response surface approximations to the nonlocal state information were developed. One neural network, used to represent the aerodynamics subspace, was a function only of the aerodynamics design variables: AR, $S_{wing}$, and fuselage size. This response surface was used to estimate a value for the aerodynamic state $L/D_{max}$ for use in the performance subspace. The weights subspace requires the complete set of design variables for determining its output states; therefore, the neural network approximation to this discipline has seven inputs mapping to the two output states. The neural network approximation to the weights subspace is used by designers in the performance subspace to estimate the merit function for the system.

The states calculated in the performance subspace, range and stall speed, depend on states from both of the other subspaces as well as a subset of the system design variables. Because the information obtained from the performance discipline is used to determine the system constraints, this response surface is used by designers in all other subspaces (as well as the system coordination problem) to approximate the constraints in their respective discipline design problems. A number of approaches could be taken to develop this response surface. It could depend upon design variables and nonlocal states, which, in turn, would depend on other design variables, or it could approximate the performance states in terms of all appropriate design variables. The latter approach was taken, and the performance subspace requires the complete set of design variables for determining its output states; therefore, the neural network approximation to this subspace has seven inputs mapping to the two output states, range and stall speed.

This simple example represents a case in which a coupled multidisciplinary design problem has been decomposed into three subspaces and the subspaces and system analysis were structured so that all state feedbacks were eliminated. Results of this application are presented next.

### Autonomous Hover-Craft Application

The AHC demonstration problem provides an increase in size (11 design variables and 13 states), and complexity (2 nested feedback loops and 4 disciplines) over the ACS problem. The AHC design models a physical system consisting of an engine, a two-bladed rotor, and airframe/payload. The imposition of a static hover condition requires that the system operate at the engine speed (rpm) that provides a thrust-to-weight ratio of unity, and this defines the design operating point. Details on the model and analyses upon which this problem was based are presented in Ref. 17.

There were 11 design variables used in this CSD application. These variables describe the physical dimensions of the rotor and the amount of fuel. Other system parameters, taken to be constant during the design process, include structural material properties, air density, the specific fuel consumption of the engine, engine weight-to-power ratio, and the weight of the payload. In addition to the operating rpm for hover, the AHC system analysis determines the required engine weight, aerodynamic loads, rotor hub stresses and tip deformation, the natural frequencies of the rotor blades in bending and torsion, fuel flow rate, tip Mach number, and the hover endurance. This system analysis was decomposed into four subspaces; aero-

dynamics, structures, propulsion and performance, and structural dynamics. The flow of information between the various discipline analyses results in several feedforward and feedback couplings. The $N^2$ diagram for the AHC problem illustrates these couplings and is shown in Fig. 4.

The coupling between the aerodynamics and structures subspaces represents a static aeroelastic problem; the deformation of the rotor is a function of the aerodynamic loads, while the loads themselves are dependent on the rotor's deformation. The coupling between the aerodynamics and performance subspaces involves the determination of the required engine size and rotor speed at the design operating condition. To solve the system of coupled, nonlinear equations that represent the system analysis, an initial estimate of engine rpm is required. The aerodynamics subspace uses this estimate of rpm to compute the aerodynamic forces. These values are fed forward into the performance subspace and used to determine the required engine power and resultant engine weight. The total thrust and system weight, which must be equal for the hover condition to be satisfied, are then considered in determining the rpm value at the next iteration. This process continued until the thrust produced and system weight were the same within a prescribed tolerance. A consistent system analysis is one for which the rotor torsional deformation and the engine rpm have converged. This problem introduced another important issue in that not all sets of design variables will result in a unique set of system states. For certain sets of design variables, the states can be multivalued or undefined. This presents considerable difficulty for any design algorithm and in the current implementation the bounds on the design variables were selected to avoid this issue.

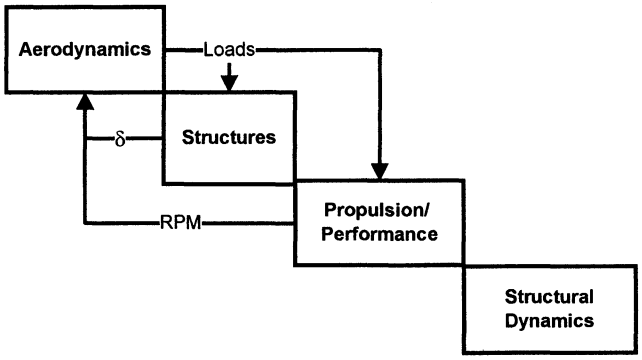


**Fig. 4 Autonomous hovercraft application.**

In the system analysis of the AHC problem, there exists feedforward/feedback information, and iteration is required. The rpm was both required as input and determined in the performance subspace. As a result, the engine rpm was a state of the system and was a function of all the system design variables. The fourth subspace, structural dynamics, is not coupled with any of the other disciplines; its input consisted solely of design variables and fixed parameters. The structural dynamics discipline calculated the first natural frequencies of the rotorcraft structure in bending and torsion.

The goal of the CSD application for this application was to minimize the weight of the vehicle subject to constraints on rotor size, shaft stress, rotor natural frequency, rotor tip Mach number, and hover-craft range. The design space defined by the parameters selected for this problem contained both a global optimum and local optimum, as shown in Table 2. These optima were determined by performing complete NAND optimizations of the system from 50 different starting points. At the global optimum, the endurance constraint is active and six of the design variables are at one of the design variable bounds listed in Table 2. Using the airfoil camber as a design variable presents an interesting problem because the empty weight varies by less than 0.01% over the entire range for this design variable. Thus, convergence at an optimum with respect to this design variable is not always observed.

In the CSD formulation, each of the four disciplines involved in the system analysis for this problem solves the system design/optimization problem using subsets of the system design vector shown in Table 3. Each subspace computes the merit and constraint functions based upon either locally computed states or approximate state information as provided by

**Table 2 AHC design variables, bounds, and optima**

| Design variable | Lower bound | Upper bound | Local optimum | Global optimum |
|---|---|---|---|---|
| Rod thickness, in. | 0.15 | 0.5 | 0.15 | 0.15 |
| Rod diameter, in. | 1.2 | 2.0 | 1.2 | 1.2 |
| Rod length, in. | 36.0 | 60.0 | 55.9 | 41.4 |
| Rotor chord, in. | 6.0 | 18.0 | 6.0 | 6.0 |
| Rotor span, in. | 24.0 | 42.0 | 42.0 | 42.0 |
| Rotor skin thickness, in. | 0.050 | 0.25 | 0.05 | 0.05 |
| Blade angle, deg | 7.5 | 15.0 | 11.1 | 10.7 |
| Airfoil $t/c$ | 0.05 | 0.20 | 0.05 | 0.05 |
| Airfoil camber, % chord | 0.0 | 5.0 | 5.0 | 2.5 |
| Rod attachment position, $x/c$ | 0.0 | 1.0 | 0.5 | 0.5 |
| $W_{\text{fuel}}$, lb | 10.0 | 50.0 | 19.0 | 19.8 |
| $W_{\text{empty}}$, lb | —— | —— | 67.9 | 67.1 |

**Table 3 AHC subspace design variable allocation**

| Design variable | Aerodynamics subspace | Structures subspace | Performance subspace | Dynamics subspace |
|---|---|---|---|---|
| Rod thickness | | X | | X |
| Rod diameter | | X | | X |
| Rod length | X | X | X | X |
| Rotor chord | X | | | |
| Rotor span | X | | | |
| Rotor skin thickness | | | | X |
| Blade angle | X | | | |
| Airfoil $t/c$ | X | | | |
| Airfoil camber, % chord | X | | | |
| Rod attachment position, $x/c$ | | X | | X |
| $W_{\text{fuel}}$ | | | X | |

the response surface approximations. Formulation of the subspace design problems is similar to the approach taken with the ACS problem. Five neural network-based response surface approximations were developed for this application, one for each of the subspaces and one for a system state, as described next. As a result of the complex couplings in this problem, the complete set of design variables were used for each response surface approximation. These response surface approximations are used to determine nonlocal state information, constraints, or the merit function during subspace design and system-level coordination.

As was seen with the ACS application, the subspace design problem for the aerodynamics subspace is completely approximate because the states on which the merit function and constraints depend are developed in other subspaces. Though it is similar to the design problem being solved in the system coordination, the difference between the system coordination problem and the aerodynamics discipline design problem is that the aerodynamics design problem utilizes only a portion of the full system design vector. In this application, it is not apparent that this subspace design problem adds anything to the process, and it was only maintained herein to simplify coding and allow for future problem extension. In certain cases there may be situations where a subspace does not directly influence merit or constraints, but it is desired that experts in that discipline have the ability to influence the design using experience, intuition, or other appropriate means.

Designers in the structures subspace have a direct impact on the merit function and two of the constraints in their subspace design problem. In addition to influencing the merit function, they have the responsibility of determining the stress constraints. The performance and structural dynamics subspace designers solve problems that are similar to that of the structural designers. In each of these disciplines, modification of geometric properties of the design results in a change in the merit function. The performance discipline designers calculate the endurance and tip Mach number constraints, whereas designers in the structural dynamics discipline determine the frequency constraints.

The frequency constraints, as well as analyses in the aerodynamics and performance disciplines, require an approximation to the motor rpm. Because rpm is not calculated within any of the disciplines, the approximation to this system state requires special consideration. Because of the way in which the AHC analysis problem is formulated, the determination of the engine rpm is performed recursively, based on a prior value of the state. As a result, the engine rpm can be affected by changes in any of the system design variables. For this reason, the engine rpm was described as a system state. To approximate the engine rpm, the influence of each design variable in the system must be incorporated. This system state was not associated exclusively with a particular discipline and was accorded its own response surface, which became part of the subspace design and system coordination problems.

The AHC implementation addressed issues regarding problem formulation and response surface approximation of subspaces involved in coupled, nonhierarchic systems.

### CSD Framework Implementation

The software for the two applications discussed earlier was developed on Unix workstations with the analysis software written in Fortan and C. The process control was accomplished using TCL/TK[18] scripts. Because these applications were not conducted by design teams, the process was performed by a single individual, and therefore, was highly automated. This somewhat limits the perceived potential of the CSD framework.

The initial step was the manual selection of the designs to populate the initial database. Once this was accomplished, the remainder of the process was automated. The system analyses were performed on each design and a design database composed of design variables and states was developed. Appropriate information was selected from the database and the neural network response surface approximation training process performed. The training data was scaled and the neural network training performed using a modified version of the NETS[19] software. The subsystem optimizations were then performed using the system-level merit function and constraints and those design variables designated as active for that subspace. The remaining design variables were fixed at their current baseline values. The subspace optimization was performed using a generalized reduced gradient (GRG)[20] algorithm with gradients developed using finite differences. Within each subspace, local states were computed using the same analysis method that was the contribution of that subspace to the system analysis. Required nonlocal states that were used for constraint and/or merit function evaluation were computed using the appropriate response surface approximations. When each of the subspace optimizers converged, a complete system analysis was performed on the resulting designs. The database was augmented with the new designs and the response surfaces were retrained. This implies that for this implementation, the only specialized expertise invoked at the subspace level was the result of using the best available local tool, i.e., the one used in the SA, and though this may not be the best way to exploit subspace experts, it was the approach used in these applications.

Once the response surface approximations were updated, a complete system-level approximation was performed again using the GRG algorithm with gradients developed using finite differences. The system analysis was fully approximate using only the neural network based approximations. The resulting design was then analyzed using the full-system analysis, a convergence check performed and if the convergence criteria was not satisfied, the design developed by the system-level optimization was selected as the new baseline design and added to the database and the process was repeated while maintaining all previous candidate designs in the database.

This automated implementation proved useful for these simple application problems. In a more realistic application, the role of the subspace expert would most likely be different and the methods they adopted for adding candidate designs to the database would be more problem specific. Lastly, the current paper does not address in detail issues related to the neural network response surfaces, but a number of these issues have been considered.[12,15–17,21–23] The neural networks can present certain benefits, but are most likely not appropriate for all applications. The type and characteristics of the most appropriate parametric response surface approximation would be a problem-dependent issue and additional research is needed in this area.

## Results

### Aircraft Concept Sizing Application

The ACS problem is a coupled problem comprised of three disciplines, seven design variables, and five states. Five trials of the CSD framework were used to study the performance of this MDO framework. All five trials were begun with an initial database of eight designs, one baseline design, and seven other designs, each with one design variable changed with respect to the baseline. In four of the five trials, the design variables were perturbed by 10% of their range. The fifth trial was conducted with a design variable perturbation of 20%. The extent of this perturbation was arbitrarily selected and would differ for other problems. Four of the five designs were started from the infeasible region and one from the feasible region in the vicinity of the local optimum discussed earlier. One of the infeasible trials began at the center of the design space. Another began with all design variables at their upper bounds, the farthest location from the global optimum in the design space considered. All five trials of the algorithm converged to

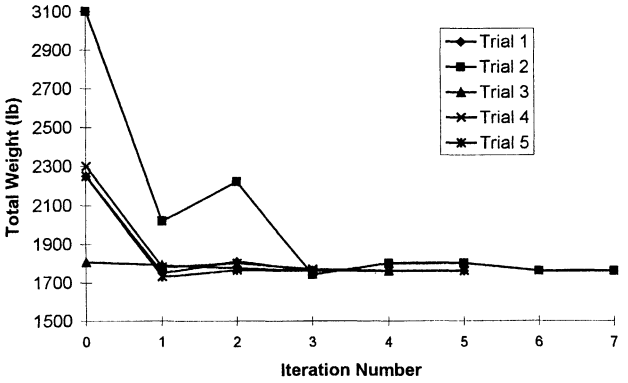


**Fig. 5 Aircraft concept sizing problem convergence histories.**

the global optimum at which five of the design variables were at lower bounds and both the range and stall speed constraints were active. The convergence histories for these five trials are shown in Fig. 5.

To measure the effectiveness of the algorithm, comparison with a NAND implementation of the problem is made. This NAND application used the full-system analysis and a GRG search with finite difference gradients. This was performed from 19 different starting points that were the 14 face centers of hypercube bounding the design space, and at the five starting points used in the CSD trials. Of the 19 starting points considered, the global optimum was identified in 63% (12) of the cases. Of the five starting points in common with the CSD trials, the NAND algorithm found the global optimum only twice. These results suggest that the CSD algorithm performs at least as well, and better in the limited number of trials, compared with the NAND procedure in locating the global optimal solution for the ACS problem.

Because an important consideration in the use of an MDO solution strategy is the cost of obtaining improved designs, it is useful to compare the number of system and discipline analyses conducted during the process. The average number of system analyses, aerodynamics analyses, and weight analyses performed during the CSD and NAND solutions are presented in Table 4.

The method labeled NAND in Table 4 represents the average number of analyses based on all 19 trials conducted. The NAND(12) method only takes into account the NAND trials that identified the global optimum. The number of analyses required for the NAND(12) trials is higher because the nonglobal optimum identified by the remaining NAND trials was closer to the starting points than the global optimum. Notice that the CSD algorithms required an order of magnitude fewer system analyses than the NAND approach. The number of discipline analyses required by CSD, however, is considerably higher than the number of discipline analyses for NAND. There are two reasons for this. The first is that the automated implementation of the CSD algorithm performed numerous optimizations (one per iteration) for each of the four disciplines. Because the CSD algorithm averaged five iterations as shown in Fig. 5, this resulted in 28 system analyses, which correspond to the eight initial designs used to populate the initial database, then four new designs added to the database per iteration (one for each of the three subspace designers and one system optimum). In this fully automated CSD application, the discipline designers solved approximate optimization problems on five separate occasions, using the same discipline analysis tools used in the system analysis. Consequently, the number of discipline analyses required during optimization in CSD is higher than in NAND in this problem.

This second factor that affects the number of discipline analyses required in CSD and NAND methods is a result of the form of the system analysis. For the ACS problem, the discipline analyses are performed only one time for each system

**Table 4 System and contributing analyses for ACS problem**

| Approach | System analyses | Aerodynamics | Weights |
|---|---|---|---|
| CSD | 28 | 989 | 13,813 |
| NAND | 229 | 229 | 229 |
| NAND (12) | 305 | 305 | 305 |

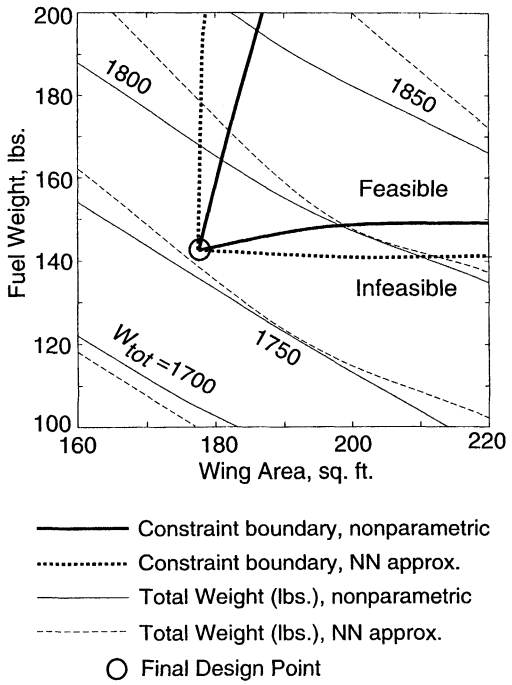


**Fig. 6 Parametric design space and constraint surfaces, ACS problem.**

analysis. In contrast, a nonhierarchic system typically requires more than one discipline analysis for each system analysis. Although the cost of performing CSD on this hierarchic problem is greater than that of NAND, the CSD algorithm has been demonstrated to be more consistent in the identification of globally optimal designs.

A concern associated with the use of the CSD algorithm is the uncertainty associated with the design identified as optimum. In certain cases, the solutions are close to the global optimum, but can converge to a suboptimal design. The suboptimal design can be slightly infeasible ($g = -0.015$, worst case) or not quite on a constraint boundary ($g = 0.001$, worst case). This occurs as a consequence of the uncertainty in the neural network approximation to the design space (states). Consider the approximation of the merit function $W_{\text{total}}$ and the constraint boundaries for the design variables $S_{\text{wing}}$ and $W_{\text{fuel}}$. Figure 6 illustrates the design space in the region around the final solution obtained in trial 1. Notice in Fig. 6 the total weight decreases with decreasing wing area and fuel weight. The neural network representation of the wing area-fuel weight cross section of the design space yields similar contours in total weight with wing area and fuel weight. Examination of the feasible region and the approximations to the constraint boundaries in Fig. 6 indicates that, at the final CSD point, the constraint boundary and approximation to the constraint boundary agree quite well. At this point, the actual range and stall velocity are 556.6 miles and 69.97 ft/s, respectively, and the approximations to range and stall velocity yield 556.2 miles and 70.0 ft/s. The span of the range values on which the neural network was trained was from 432.2 to 966.6 miles. The error in range at the final CSD design is 0.08% of its nominal value. The stall velocities at the points in the training database were in the range of 57.80–75.24 ft/s, corresponding to an approximation error at the final point of 0.17%. The final

**Table 5 AHC initial trial results**

| Trial | Rod length, in. | Blade angle, deg | Rod position, $x/c$ | Camber, % | Fuel weight, lb | Endurance constraint | Empty weight, lb |
|---|---|---|---|---|---|---|---|
| CSD 1 | 45.8 | 11.4 | 1.0 | 5.0 | 18.6 | −0.043 | 66.9 |
| CSD 2 | 36.0 | 9.9 | 0.41 | 0.0 | 19.3 | −0.050 | 66.8 |
| CSD 3 | 36.0 | 14.5 | 0.0 | 0.0 | 20.8 | −0.019 | 69.8 |
| Optimum | 41.4 | 10.7 | 0.50 | 2.5 | 19.8 | 0.000 | 67.1 |
| Reason | ——[a] | ——[a] | ——[b] | ——[b] | ——[a] | —— | —— |

[a]Uncertainty in response surface approximation.
[b]Small influence of design variable.

**Table 6 AHC reduced design variable range results**

| Trial (initial database size) | Rod length, in. | Blade angle, deg | Rod position, $x/c$ | Camber, % | Fuel weight, lb | Empty weight, lb |
|---|---|---|---|---|---|---|
| CSD 4 (12) | 40.0 | 12.0 | 0.50 | 2.4 | 19.8 | 67.9 |
| CSD 5 (12) | 40.0 | 12.0 | 0.49 | 2.5 | 19.2 | 68.9 |
| CSD 6 (8) | 40.0 | 12.0 | 0.50 | 2.5 | 20.9 | 69.6 |
| Optimum | 41.4 | 10.7 | 0.50 | 2.5 | 19.8 | 67.1 |

CSD point (from trial 1) is slightly infeasible ($g_1 = -0.015$). Convergence to an infeasible point in this instance is a consequence of the optimization algorithm and the shape of the constraint boundary approximation. Although the constraint boundary approximation in Fig. 6 represents the general characteristics of the actual constraint boundary, in the vicinity of the baseline design (the circle in Fig. 6) the shape of the approximation limits the optimizer's ability to identify a feasible point. At the baseline point the gradients of the constraint boundary approximation in both the $S_{\text{wing}}$ and $W_{\text{fuel}}$ directions are zero and all other design variables are at their lower bounds. Consequently, a new search direction that results in a feasible design cannot be formed. Thus, the optimization procedure causes the CSD algorithm to terminate at a design that is slightly infeasible. In the current implementation, the responsibility for design convergence and feasibility rests with the optimization routine. In a more interactive design environment, the constraint boundary could be easily explored (because it is approximated fully), and a new baseline point for the system coordination identified.

### AHC Application

The AHC problem is a coupled problem involving four design subspaces, 11 design variables, and 18 approximated states. The merit function for this problem is monotonic in all but two design variables; consequently, one would expect the solution to this problem to occur on constraints or at design variable bounds as it does. A fuel weight of 19.8 lb is needed to meet the endurance constraint at the global optimum. Three initial trials of the CSD algorithm were performed. All three trials were begun with an initial database of 12 points, one baseline design and 11 other designs, each with one design variable changed with respect to the baseline. In all three trials, the design variables were perturbed by an arbitrarily selected 10% of their full range about the baseline. The first starting point had all design variables at their upper bounds. This starting point placed one of the design variables, wingspan, at its optimum value and the rest as far from their optimal values as possible. The starting point in trial 2 had each design variable in the middle of its range. The final starting point, trial 3, placed all of the design variables at the boundary that was as far as possible from their optimal values.

An issue of concern in the results of the CSD application to the AHC problem is the ability to identify optimal designs. For the three trials, all solutions obtained were slightly infeasible. In all three cases, the endurance constraint was violated. The final values for the design variables that were not at the value corresponding to the optimal, the violated constraint, and the merit function values for each of the three CSD cases are presented in Table 5. The reason for the suboptimal designs appears to be problem specific. Recall that the entire design database is used to construct approximations of the system states at all points in the design process. Because of this, a global approximation to the system states is obtained. In the ACS application the error in the neural network response surface approximations was small compared with the variation in the merit function and constraints over the range of a particular design variable. For the AHC problem a 1% neural network training tolerance corresponds to a change in empty weight (the merit function) of about 2.4 lb. In this example, however, a 1% training error (2.4 lb) is on the same order of magnitude as the variation of $W_{\text{empty}}$ over the entire design variable range for the rod length. Hence, for the design variables for which the change in the merit function over the design variable range is small with respect to the entire range of the merit function, the response surface approximation can be ambiguous, and the design identified by CSD may be suboptimal. This is the situation that occurred in the AHC problem.

In an attempt to verify this observation, another set of trials (4–6) were conducted, and the comparable results are presented in Table 6, which includes the number of initial designs in the database. In these trials, the range on the design variables was reduced about the design point, identified as the optimum in the earlier trials. In this case, the range on empty weight was reduced so that a 1% training tolerance corresponded to approximately 0.4 lb. As can be seen in Table 6, the improved response surface approximations resulted in designs more proximate to the global optimum. Improved approximations in the region of the optimum were achieved by excluding designs from the database that were far from the final design point. Improved convergence to the global optimum was obtained at the expense of a more global approximation. This issue can be problem dependent as some design studies are intended to locate the optimum design and others to perform what if studies over a wider range of the design space. Other examples have shown that improved designs can be identified, even if the approximations are somewhat inaccurate locally, but reflect the overall trends in the design space.

In addition to being able to obtain improved designs via the CSD algorithm, it is also desirable to reduce the number of analyses (both system and subspace). Again the NAND procedure was the benchmark. Fifty NAND trials were conducted for the AHC problem. Of the 50 trials, 39 reached the global optimum (78%), and the remainder converged to the local optimum. The number of system and subspace analyses required to identify an optimal solution depends on the starting point of the NAND algorithm; hence, there was a wide range in the number of system and subspace analyses performed using this

**Table 7 System and contributing analyses for AHC problem**

| | System analyses | | | Aerodynamics | | |
|---|---|---|---|---|---|---|
| Trial | Mean | High | Low | Mean | High | Low |
| NAND | 3018 | 51,461 | 183 | 33,382 | 583,129 | 1948 |
| CSD 1 | 72 | —— | —— | 2846 | —— | —— |
| CSD 2 | 72 | —— | —— | 2908 | —— | —— |
| CSD 3 | 107 | —— | —— | 4911 | —— | —— |
| CSD 4 | 67 | —— | —— | 1276 | —— | —— |
| CSD 5 | 27 | —— | —— | 737 | —— | —— |
| CSD 6 | 33 | —— | —— | 798 | —— | —— |

method. The average and upper and lower bounds on the number of analyses required to identify optimal designs using the NAND algorithm are presented in Table 7.

The initial three CSD trials were able to move the baseline design to the neighborhood of the global optimal solution (in all three trials), utilizing 58% of the lowest NAND number of system analyses. Comparing the worst of the three CSD trials with the average number of system analyses required using NAND, the CSD trial utilized 3.5% of the system analyses to converge. Recall that this CSD trial was initiated at the point in the design space that was farthest from the global optimum.

The current automated implementation of the CSD algorithm relies on subspace designers, who utilize their discipline tools in conjunction with neural network approximations to nonlocal states during subspace design. Thus, the number of subspace analyses per system analysis performed in the course of the CSD algorithm is higher than that of NAND. The NAND procedure required an average of 11.1 discipline analyses for each system analysis for a total of 33,382 analyses in each of the coupled disciplines (aerodynamics, structures, and performance) and the same number of analyses of the structural dynamics discipline as system analyses (3018). In contrast, the number of discipline analyses performed in the CSD algorithm for a given discipline depends on the number of design variables allocated to that discipline, the baseline design point (optimization starting point), the current design space approximation, and other factors. This results in variability in the number of discipline analyses performed between coupled disciplines and variability for a single discipline between CSD trials. In the six CSD trials, the algorithm required fewer discipline analyses for each discipline than the average number of discipline analyses required using NAND. Again, the location of the CSD starting points for these trials was far from the optimal solution.

The intent of the AHC problem was to provide a demonstration of the CSD algorithm on a problem with more MDO features (nonhierarchic with multiple couplings). There have been a number of other applications of this approach to a variety of other problems. These studies have addressed a variety of other issues in more detail including neural network training, the use of mixed continuous and discrete design variables, the effects of changing requirements during the design process, and the variations in the methods used during subspace design.[21–27] Each of these issues are of importance in the design of many engineering systems.

## Conclusions

A framework has been proposed to allow for the multidisciplinary design of coupled, nonhierarchic systems. This approach, which is referred to as the concurrent subspace design, exploits the ability to decompose the model-based analysis of a coupled system into subspaces or contributing disciplines. These subspaces are defined in terms of the information they contribute to the characterization of the complete system. The coupling of the subspaces is based on the information exchanged between them. The complexity of the information coupling influences the cost and characteristics of the system analysis. The primary goal of this approach has been to reduce the number of complete system analyses required to improve or optimize the design.

The CSD framework is based on the assumption that design decisions can occur at the subspace level if the subspace experts can approximate the information required from other subspaces as they attempt to improve the merit of the system and meet the system constraints. In CSD, this information is developed using a set of response surface approximations based on artificial neural networks. Unlike local sensitivity information, these approximations provide a parametric representation of the complete design space and allow for large excursions within the design space. Because the subspace design can allow for sharing of design variables, coordination of the subspace design decisions is required and achieved by the solution of a fully approximate system-level optimization problem using the complete set of design variables.

Two simple applications were presented to illustrate problem formulation and framework implementation. In these applications, subspaces were defined using traditional disciplines or groups of disciplines and included information coupling and feedback. The subspace definition significantly influences the way in which the response surface approximations are formed. In the formulation presented herein, every design variable whose variation would influence a particular system state was required as input to the neural network response surface approximation. This approach to formulating response surfaces would appear to limit the approach to applications with relatively few global design variables. Most benefit would likely be achieved in applications where the number of design variables is small (<50), but the cost of performing a system analysis is high. In each application considered, the CSD framework provided optimum or near optimum designs with the number of system analyses significantly reduced from more traditional complete system optimization approaches. The use of the GRG search using finite difference gradients, coupled with the full-fidelity analysis tool within the subspace for defining the candidate designs in the subspace optimization phase would likely be impractical in most applications. It would need to be replaced by more economical schemes for selecting candidate designs to augment the design database, and recent efforts have suggested that alternative approaches are feasible.

There appears to be good reason to consider the use of CSD for cases in which a rich database might exist as a result of earlier design studies or problems in which there are multiple merit functions or changes in the form of the merit or design requirements can occur during the process. These situations and those that require the use of mixed continuous and discrete design variables are of significant interest in many practical design problems and they should provide challenging applications for CSD in the future.

## Acknowledgments

This work was supported in part by NASA, Langley Research Center, Grant NAG-1-1561, J. Sobieszczanski-Sobieski, Project Monitor. The authors also wish to acknowledge the support of their colleagues and collaborators, J. Renaud, J. Brockman, and B. Wujek, for their contributions to this work.

## References

[1]Balling, R. J., and Sobieszczanski-Sobieski, J., "Optimization of Coupled Systems: A Critical Overview of Approaches," *Proceedings of the AIAA/NASA/USAF/ISSMO 5th Symposium on Multidisciplinary Analysis and Optimization* (Panama City, FL), AIAA, Washington, DC, 1994, pp. 697–707.

[2]Raymer, D. P., *Aircraft Design: A Conceptual Approach*, AIAA, Washington, DC, 1989, pp. 11–32, 101–116, 509–526.

[3]Rogers, J. L., "DeMAID—A Design Manager's Aide for Intelligent Decomposition, User's Guide," NASA TM-101575, March 1989.

[4]Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," AIAA Paper 96-0711, Jan. 1996.

[5]Sobieszczanski-Sobieski, J., "Multidisciplinary Design Optimization: An Emerging, New Engineering Discipline," *Advances in Structural Optimization*, edited by J. Herskovits, Kluwer Academic, Norewell, MA, 1993, pp. 483–496.

[6]Johnson, E., and Venkayya, V., "Automated Structural Optimization System (ASTROS), Theoretical Manual," Vol. 1, U.S. Air Force Wright Aeronautical Labs., TR-88-3028, 1988.

[7]Sobieszczanski-Sobieski, J., "Optimization by Decomposition: A Step from Hierarchic to NonHierarchic Systems," NASA CP-3031, Sept. 1988.

[8]Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," *Proceedings of the AIAA/NASA/USAF/ISSMO 5th Symposium on Multidisciplinary Analysis and Optimization* (Panama City, FL), AIAA, Washington, DC, 1994, pp. 697–707.

[9]Sobieszczanski-Sobieski, J., "On the Sensitivity of Complex, Internally Coupled Systems," NASA TM-107622, Jan. 1988.

[10]Renaud, J. E., and Gabriel, G. A., "Improved Coordination in Nonhierarchic System Optimization," *AIAA Journal*, Vol. 31, No. 12, 1993, pp. 2367–2373.

[11]Wujek, B. A., and Renaud, J. E., "Design Driven Concurrent Optimization in System Design Problems Using Second Order Sensitivities," *Proceedings of the AIAA/NASA/USAF/ISSMO 5th Symposium on Multidisciplinary Analysis and Optimization* (Panama City, FL), AIAA, Washington, DC, 1994.

[12]Swift, R., and Batill, S. M., "Application of Neural Networks to Preliminary Structural Design," AIAA Paper 91-1038, April 1991.

[13]Swift, R. A., and Batill, S. M., "Damage Tolerant Structural Design Using Neural Networks," AIAA Paper 92-1097, Feb. 1992.

[14]Batill, S. M., and Swift, R. A., "Structural Design Space Definition Using Neural Networks and a Reduced Knowledge Base," AIAA Paper 93-1034, Feb. 1993.

[15]Batill, S. M., and Swift, R. A., "Preliminary Structural Design—Defining the Design Space," U.S. Air Force Wright Lab., TR-3004, Wright–Patterson AFB, OH, Feb. 1993.

[16]Hajela, P., and Berke, L., "Neurobiological Models in Structural Analysis and Design," *AIAA 31st Structures, Structural Dynamics, and Materials Conference* (Long Beach, CA), AIAA, Washington, DC, 1990, p. 345.

[17]Sellar, R. S., "Multidisciplinary Design Using Artificial Neural Networks for Discipline Coordination and System Optimization," Ph.D. Dissertation, Univ. of Notre Dame, Notre Dame, IN, April 1997.

[18]Ousterhout, J. K., *TCL and the TK Toolkit*, Addison–Wesley, Reading, MA, 1994, pp. 1–24.

[19]Baffes, P. T., "NETS 2.0 Users Guide," NASA Johnson Space Center, Sept. 1989.

[20]Gabriele, G., and Beltracchi, T., "OPT3.2: A FORTRAN Implementation of the Generalized Reduced Gradient Method, User's Manual," Dept. of Mechanical and Aerospace Engineering and Mechanics, Rensselaer Polytechnic Univ., New York, 1988.

[21]Carpenter, W., and Barthelemy, J.-F. "Some Common Misconceptions About Neural Networks," *Journal of Computing in Civil Engineering*, Vol. 8, No. 3, 1994, pp. 345–359.

[22]Sellar, R. S. and Batill, S. M., "Concurrent Subspace Optimization Using Gradient-Enhanced Neural Network Approximations," AIAA Paper 96-4019, Sept. 1996.

[23]Stelmack, M., and Batill, S. M., "Neural Network Approximations of Mixed Continuous/Discrete Systems in Multidisciplinary Design," AIAA Paper 98-0916, Jan. 1998.

[24]Sellar, R. S., Batill, S. M., and Renaud, J. E., "Response Surface Based, Concurrent Subspace Optimization for Multidisciplinary System Design," AIAA Paper 96-0714, Jan. 1996.

[25]Sellar, R. S., Stelmack, M., Batill, S. M., and Renaud, J. E., "Response Surface Approximations for Discipline Coordination in Multidisciplinary Design Optimization," AIAA Paper 96-1383, April 1996.

[26]Stelmack, M., and Batill, S. M., "Concurrent Subspace Optimization of Mixed Continuous/Discrete Systems," AIAA Paper 97-1229, April 1997.

[27]Stelmack, M., Batill, S. M., Beck, B., and Flask, D., "Application of the Concurrent Subspace Design Framework to Aircraft Brake Component Design Optimization," AIAA Paper 98-2033, April 1998.