AIAA 92-4791

The Suitability of Selected Multidisciplinary Design and Optimization Techniques to Conceptual Aerospace Vehicle Design

J. Olds
N.C. State University
Raleigh, NC

4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization

September 21-23, 1992
Cleveland, OH

The Suitability of Selected Multidisciplinary Design and Optimization Techniques to Conceptual Aerospace Vehicle Design

John R. Olds*
N. C. State University, Raleigh, NC

Abstract

Aerospace vehicle conceptual design is dominated by interactions among various traditional engineering disciplines. Aerodynamics, propulsion, performance, weights, sizing, and others are usually highly coupled, and complete vehicle analysis requires an iterative process with efficient methods of communication among the disciplines. Progress to computerize the analysis process has been fast in recent years, producing analysis tools such as NASA-Langley's AVID and EASIE. Given a configuration, the capability exists to quickly analyze it in order to determine its overall characteristics and performance.

However, the vehicle designer/ integrator still largely depends on intuition to make systems level changes to the configuration and components in order to improve or optimize the overall design. "What if" studies are typically performed by perturbing the design variables one at a time in an attempt to locate a better design. A complete reanalysis of the entire system is then required for each variable change. This method is a time consuming process that may or may not lead to a more desirable vehicle design.

Several mathematically based design techniques have recently emerged that could help the system designer make necessary improvements. These new methods serve to bridge the gap between analysis and design. This paper attempts to give a brief overview of four such techniques, system decomposition, sensitivity analysis, Taguchi methods, and for comparison, classical optimization. References to examples of successful uses of each technique are provided. The goal of this paper is to assess the pros and cons of each technique and their applicability to aerospace vehicle conceptual design.

* - graduate student, mech. and aero eng. dept.

Introduction

Efforts to improve the analysis techniques used to rapidly analyze an aerospace vehicle concept (i.e. determine its performance, aerodynamic coefficients, TPS requirements, etc. for a given set of inputs) have produced great strides in efficient systems integration and data interchange capabilities. Recent experience has taught that in order to maintain a sufficient level of analysis detail, it is desirable to make use of the existing analysis tools of the disciplinary experts rather than writing a new, all-in-one analysis code that may easily become outdated and unwieldy. Therefore, executive systems have been created to link the outputs of one computational analysis code to the inputs of others, to maintain a database of design variables, and to keep an audit trail of design changes (fig. 1).

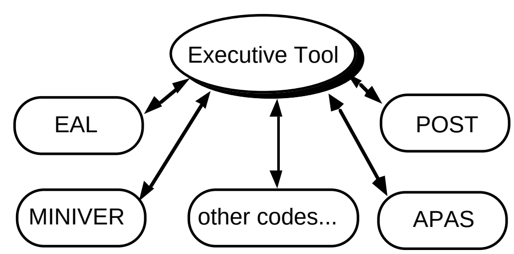

Figure 1 - Integrated Analysis with Existing Codes

For example, at NASA - Langley Research Center, the Vehicle Analysis Branch has developed an executive tool called EASIE - Environment for Application Software Integration and Execution[1]. EASIE provides a loose data interfacing framework within which existing (sometimes commercially developed) computer analysis tools can be linked. Inputs and outputs of different analysis codes are exchanged electronically, thereby increasing the efficiency and speed of a complex analysis task that would usually require manual data exchange between various disciplinary experts. EASIE also provides a central database of design variables It does not, however, provide any "suggestions" for improvements or ways to meet all of the system constraints. For that, the systems designer must rely on intuition and experience. EASIE is an evolution of an earlier system at NASA called AVID - Aerospace Vehicle Interactive Design[2]. Like EASIE, AVID provided an interfacing framework for data exchange between various disciplinary analysis codes.

In order to truly take advantage of the increased ability to perform quick turn around aerospace vehicle analysis, it is necessary to develop a systems level design methodology that will provide necessary information to enable a vehicle designer to make changes to a vehicle design that will improve its objective function and satisfy all constraints. The objective function may be a design goal like reduced weight, increased payload, or reduced system cost. In a typical design problem, all of the traditional disciplines are highly coupled (e.g. performance trajectories depend on weights, and weights depend on loads which, in turn, are dependent on performance trajectories). Answers to most "what if" questions are far from intuitive. Reference 3 gives the following example, "...making a wing airfoil thicker will tend to lighten the wing structure, but it will also increase the wave drag of a supersonic aircraft. The net effect on performance may be positive, negligible, or negative, depending on overall system [interactions]".

Parametric studies can provide a broad perspective of the effect of changing selected variables (while others are held constant). However, parametric studies tend to be slow, and the results depend strongly on the range over which the variables were changed, which variables were held constant, and the values at which they were held. A more desirable methodology would provide a fast and accurate assessment of the influences that the different independent design variables have on the objective function and constraints of the current configuration. Additionally, the new methodology would provide feedback to the designer enabling him or her to make necessary improvements to the configuration and quickly see the results of the changes. If desired, the new methodology should be able to work with a numerical optimizer to enable rapid design convergence.

Four design techniques are candidates to fill this design methodology requirement - classical optimization (presented primarily for comparison purposes), system decomposition, sensitivity analysis, and Taguchi methods. A short review of each method is given here with the goal of assessing the suitability of each for multidisciplinary aerospace vehicle conceptual design. A brief mathematical discussion, pros and cons, and an overview of a previous application example is included for each.

Nomenclature

Classical Optimization

α - gradient method step size
f - system objective function
∇f - objective function gradient
g - system design constraints
OWRA - oblique wing research aircraft
SAS - stability augmentation system
x_i - independent design input variable
X - vector of design variables
y_i - individual analysis output variable

System Decomposition

COFS - control of flexible structures experiment
f - system objective function
f_i - subproblem objective function

Sensitivity Analysis

BFL - balanced field length (take-off distance)
GSE - global sensitivity equations
S_c - canard surface area
S_t - tail surface area
S_w - wing surface area
SSA - system sensitivity analysis
SSD - system sensitivity derivatives
TOGW - aircraft take-off gross weight
X_c - canard distance from aircraft nose
X_t - tail distance from aircraft nose
X_w - wing distance from aircraft nose
X - vector of system inputs
Y - vector of system outputs
$\mathbf{Y}_i$ - disciplinary partitions of output vector

Taguchi Methods

ANOM - analysis of the mean technique
ANOVA - analysis of variance technique
AR - engine area ratio
contrast - transformed vers. of an orthogonal array
CONSIZ - configuration sizing and weights code
DOF - degrees of freedom

I_{sp}	- engine specific impulse
MR	- engine mixture ratio
M_{tr}	- Mach number of mixture ratio transition
n	- number of rows of the noise array
P_c	- engine chamber pressure
POST	- program to optimize simulated traj.
S/N	- signal-to-noise ratio
SSTO	- single stage to orbit
T/W_o	- initial vehicle thrust to weight
w_i	- weighted values used in the contrast array
y_i	- objective function for each experiment

Classical Optimization

Classical optimization techniques can be readily applied to some of the more manageable problems of aerospace vehicle conceptual design. The process first requires that the problem be placed in standard form (discussed below). Given a starting, non-optimum design, the method then steps from one design to the next until an optimum design is found that meets all constraints and minimizes the objective function. The method can take advantage of a variety of available non-linear optimization numerical methods such as variable metric, steepest descent, and several non-gradient techniques like Powell's method and random walk[4]. The multidisciplinary nature of the design problem will usually necessitate an iterative approach - sometimes even within each step.

The standard form of a classical optimization problem has an objective function and a list of constraints. A composite objective function representing a weighted assessment of the goals of each of the disciplines is written in the form:

$$\text{minimize } f(y_1, y_2, ..., y_i, ...) \qquad (1)$$

and the corresponding set of constraints written in the standard form are:

$$g_i\,(y_1, y_2, ..., y_i, ...) \leq 0 \qquad (2)$$

For example, if f was vehicle weight, the y's might be subsystem weights. The overall objective function is a function of the lower level outputs. A typical constraint may be minimum deliverable payload. The analysis outputs, y_i, are functions of the independent design input variables, x_i. Therefore, the goal of the design process is to find the design variables, x_i, which will minimize the objective function while satisfying all of the constraints. Continuous variables are easily handled by the method. Discrete variables, like number of engines or number of boosters, are more difficult to handle, but techniques exist to accommodate them.

Figure 2 shows the graphical depiction of a design space with two independent variables. The classical optimization method employs a repetitive convergence method by first assessing the system objective functions and constraints, calculating gradients, updating the design variables, and then reevaluating the new objective function and constraints. Using this stepping technique, an optimum, feasible solution is eventually found. Because the method steps from one solution to a better one by changing design variables appropriately, it is not necessary to predetermine a limiting range of each variable like a parametric study would do. The entire design space can be used if needed because there are no artificial limits.

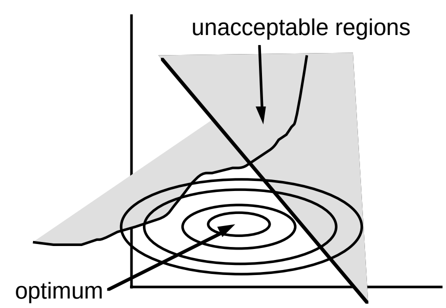

Figure 2 - Design Space with Constraints

The constraints in classical optimization can be treated in a variety of ways. For linear problems, the Simplex method from linear programming can be used[4]. The Simplex method uses the fact that the optimum solution to a linear problem (linear objective and linear constraints) will be at the intersection of two constraints. Those constraints are given the designation "active", and the "≤" is replaced by "=" in equation (2). For more typical non-linear problems, the constraints can be represented by penalty functions that treat the constraints, not as on-off step functions, but as steeply sloping functions beginning at the point where the constraint is "just satisfied" and increasing as the design moves away from the feasible region. The penalty functions are zero inside the feasible region. The penalty function is then added (for a minimization problem) to the overall objective function. In this way, the objective is "penalized" for being outside the feasible region, and gradient methods will lead the design away from the penalties and toward a feasible solution.

Gradient methods used in classical optimization, as the name implies, use the gradient of the objective function, ∇f, to perform the minimization. They start with a given set of design variables and then numerically, or analytically, determine the derivative of the objective function in each of the design variables directions, i.e. the gradient.

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_i}, \ldots\right) \qquad (3)$$

The simplest gradient method, steepest descent, then uses the fact that the negative of the gradient lies in the direction that most improves the objective function (for a minimization problem). Therefore, the vector of design variables, represented by **X**, is changed in that direction.

$$\mathbf{X}_{new} = \mathbf{X}_{old} - \alpha * \nabla f \qquad (4)$$

In equation (4), α is a scalar that varies the magnitude of the step. Once the gradient direction is determined, α is started at a small value, an intermediate value of f is determined, and α is systematically increased until the value of the intermediate f is no longer an improvement over the previous step. In other words, the current gradient direction is followed until it is "played out". In practice, maximum move limits are sometimes established to keep the optimizer from taking too large of a step in a non-linear problem. Once a best α is determined for the current gradient direction, the design variables, **X**, are updated and a new gradient direction is calculated. From here, the process is repeated until the problem converges. Other numerical optimization techniques may use different methods to update the design variables, but almost all use a stepping scheme.

The minimum (or maximum) is found when the derivatives of all the variables are equal to 0,

$$\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = \frac{\partial f}{\partial x_i} = 0 \qquad (5)$$

A minimum is, therefore, a point where changing any design variable will result in an increase in the objective function (the increase in the objective function may be the result of a penalty from a violated constraint). It is possible that the optimization process may find a local minimum which is not the global minimum. Techniques exist to solve this problem, but most involve starting the design from a new initial condition.

<u>Pros and Cons</u>

The classical optimization technique depends heavily on the ability to quickly evaluate the objective function and constraints at each iteration (and several times within each iteration to evaluate the derivatives). The multidisciplinary nature of most aerospace vehicle designs makes this requirement very difficult to achieve. The current philosophy of distributed experts and existing analysis codes would require that each discipline perform an analysis for each iteration of the solution. If the system is coupled, the solution process becomes even more complex and time consuming. Discrete variables are more difficult to accommodate in this method than in parametric studies. Additionally, the objective function and constraints may become very difficult to formulate in a standard form. Therefore, the classical optimization technique should only be applied to a limited class of problems.

If the design problem can either be limited in scope or approximations can be made to simplify the analysis equations, classical optimization becomes a viable method. A simple objective function and set of constraints must be written in order to allow fast evaluation. While the problem can still be multidisciplinary, complex computer codes for detailed aerodynamics, propulsion, controls, and structures analysis are generally discarded in favor of approximate methods - simple algebraic equations in most cases. Applied to a suitable problem, classical optimization provides a numerically optimum solution (limited only by the accuracy of the model, not the method), a design that meets all constraints, and a method that doesn't require the designer to place predetermined limits design variable ranges.

<u>Example Application</u>

In reference 5, Morris and Kroo of Stanford

University apply the methods of classical optimization to two different multidisciplinary design problems - control of a tailless flying wing aircraft (controls, structures, and aerodynamics) and to an oblique wing aircraft (controls, geometry, and aerodynamics). In each case, a multidisciplinary objective function with goals from each discipline is formulated, constraints are established, and an iterative analysis loop using an integrated system of relatively simple analysis tools is used to produce significant handling improvements.

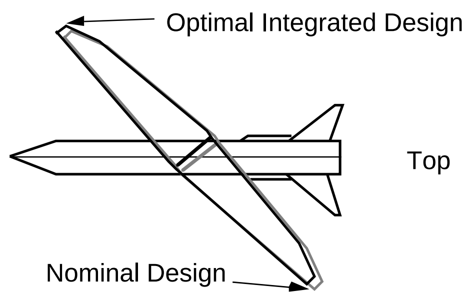

Fig. 3a - Oblique Wing Aircraft Redesign (Ref. 5)

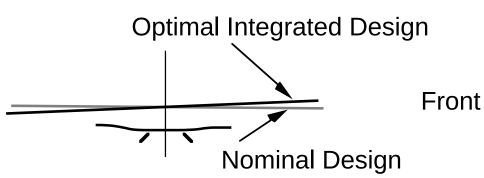

Fig. 3b - Oblique Wing Aircraft Redesign (Ref. 5)

Taking the case of the oblique wing aircraft (as shown in Figures 3a and 3b taken directly from reference 5), the objective function was to improve the aircraft dynamic handling characteristics as much as possible by attempting to decouple the lateral and longitudinal equations of motion. Before optimization, the oblique wing research aircraft (OWRA) design experienced significant lateral acceleration and roll during a pull-up pitch maneuver. The design was optimized by treating the position and orientation of the oblique wing on the fuselage, the wing dihedral, and the stability augmentation system (SAS) feedback gains as design variables. Constraints in the form of penalty functions were placed on the ability of the redesigned aircraft to trim at flight conditions and the limits of movement of the control surfaces. An iterative analysis technique was created using a control system synthesis tool coupled with a vortex paneling method for aerodynamic analysis. Mass properties and moments of inertia were also calculated for each iteration. Penalty functions were used to model the constraints, and a quasi-newtonian optimizer provided the design variable changes. See figure 4 for a flow chart of how the method was applied.

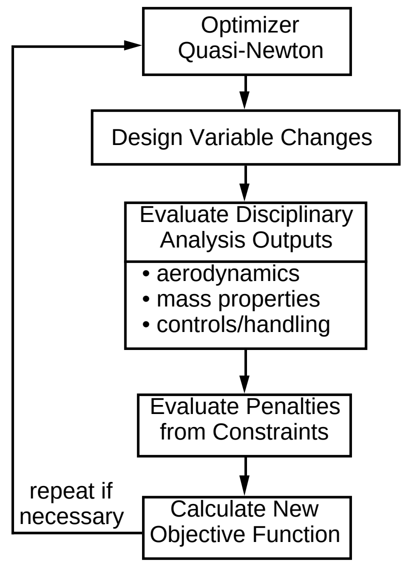

Figure 4 - Method Application Flowchart (Ref. 5)

At the optimal solution, Morris and Kroo moved the wing slightly (changing its bank angle relative to the fuselage) compared to the nominal wing location on the OWRA. The result was a significant improvement in dynamic handling. For a 4-g pull-up maneuver, the peak lateral acceleration was reduced from .31-g to .04-g, and the peak roll angle was reduced from 19° to 7.5°[5] (see figures 5 and 6 for results taken from reference 5).

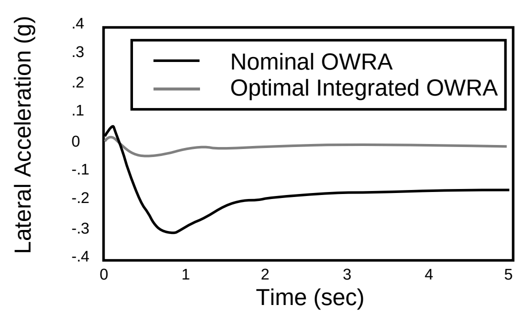

Figure 5 - Reduced Lateral Accel. Results (Ref. 5)

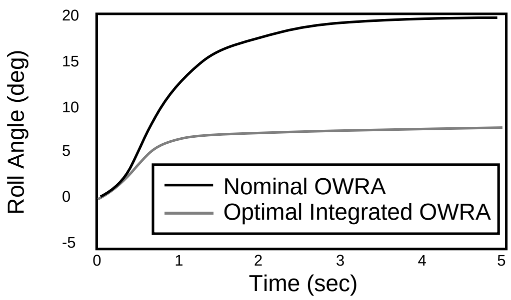

Figure 6 - Reduced Roll Results (Ref. 5)

For another example of classical optimization, see reference 6 in which Kroo and Gallman apply a similar technique to the optimization of a joined wing aircraft.

System Decomposition

If a system consists of several coupled disciplines or tasks, it may be possible to organize the system into a top down hierarchy of smaller subproblems or combinations of subproblems (figure 7). This process of decomposing the coupled system leads to a simpler set of subproblems that can be optimized in a one-at-a-time manner rather than the all-at-once manner employed by classical optimization. In a sense, system decomposition enables the extension of the ideas of classical optimization to larger coupled problems.

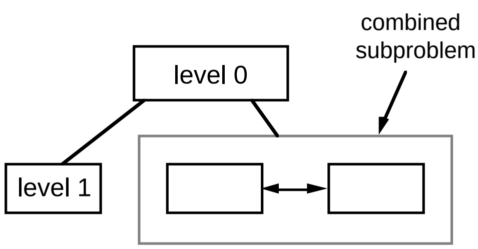

Fig. 7 - Complex System as a Hierarchical Structure

Because multi-level decomposition generates a series of subproblems, it lends itself well to the idea of retaining the existing tools of the disciplinary experts and using them to provide the required level of analysis detail in an overall design optimization problem. Once the system is decomposed, the subproblems of the hierarchical tree can be treated as "black boxes" providing outputs to and receiving inputs from other contributing subproblems. Optimization of the overall design is accomplished by top down optimization of the elements of the decomposed structure. Compared to classical optimization, fewer simplifications to the analysis are required because the existing, detailed design programs of the disciplinary experts are retained.

The setup of a decomposition problem involves describing the overall system as individual subsystems (modules), their output variables, and their input requirements. For example, thermal protection analysis may be a module. It would provide TPS type, thickness, and weight as outputs and require aerodynamic heat loads and structural backface temperature limits as inputs. Once the entire network is created, the modules are organized in a manner that reduces feedback (iteration) from lower to upper levels and creates a logical hierarchical structure. Some modules may be so coupled that they are impossible to break apart. They may instead be combined into a larger subproblem (circuit) within which iteration may occur. Decomposition may lead to aerodynamic analysis being performed before TPS analysis, and the propulsion and propellent tank analysis may be combined into a new "circuit", for example.

At NASA - Langley Research Center, a knowledge based tool called DeMaid - Design Manager's Aid for Intelligent Decomposition[7] was created in order to automate the process of decomposition (figure 8). In the simple four module N by N graph example shown, module 2 provides output to modules 1 and 3, and modules 3 and 4 provide outputs to each other. DeMaid uses knowledge base rules in order to transform the coupled, complex design problem into a more manageable hierarchical structure of subproblems. The reorganized system recommends analysis in module 2 first (it depends on input from no other module) and combines modules 3

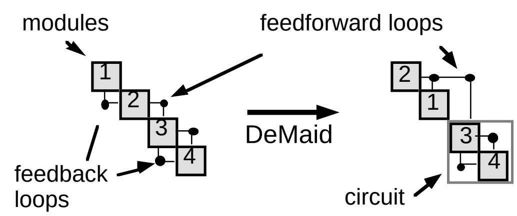

Figure 8 - Decomposition Reduces Feedback

and 4 into a new circuit. A top down analysis is now possible because no feedback loops exist.

Assuming that the system can be decomposed (some systems may be too highly coupled to create a hierarchical structure), a structured process can be utilized to optimize the individual subproblems so that the optimum solution of the subproblems is the optimum solution of the entire system[8]. This process is known as coordination.

As in the case of classical optimization, the overall objective function can be written as a function of the outputs of the individual disciplines (modules).

$$\text{minimize } f(y_1, y_2, ..., y_i, ...) \qquad (6)$$

In the simplest decomposition problem, the outputs of each of the modules enter only into the calculation of the objective function and not into the outputs of any of the other modules (a block diagonal dependency matrix system). As an example of coordination, assume this case existed. It would enable the objective function to be split into a series of smaller independent objective functions so that:

$$\sum f_i(y_i) = f(y_1, y_2, ..., y_i, ...) \qquad (7)$$

and the objective function of the i-th subproblem will become:

$$\text{minimize } f_i(y_i) \qquad (8)$$

The system constraints can also be broken into smaller, subproblem level constraints. In a simplified example, if the system objective function is to minimize vehicle weight, the TPS subproblem objective may be to minimize TPS thickness. Other subproblems would have their own objective functions so that when each is optimized separately, the result will be the lightest overall vehicle.

Having decomposed the system into a hierarchical structure and rewritten the systems level objective function and constraints into subsystem level equations, the design process proceeds from the top of the hierarchical tree to the lower levels. Recall that higher level subproblems are analyzed before lower level subproblems because the higher level outputs (feedforward data) are required as inputs to lower level analyses. The individual optimizations of the subproblems can be solved in a variety of ways, including some of the non-linear, numerical optimization methods like steepest descent discussed in the classical optimization section. For smaller subproblems, designer experience may be sufficient to find an appropriate solution.

<u>Pros and Cons</u>

System decomposition enables a designer to extend the ideas of classical optimization to larger and more coupled problems. However, if a problem is very tightly coupled, it may be impossible to decompose the problem into a simple set of subproblems that can be handled by classical optimization. Simplifications are often made to reduce the system coupling, that is, some of the weaker dependencies are often neglected. Alternatively, if the overall design can be broken down into 2 or more highly coupled subproblems, the system sensitivity analysis technique (discussed later) can be used on each of the subproblems separately, and then the problems can be recombined using coordination thereby saving time and effort.

Because of the branching nature of the hierarchical tree, unrelated lower level subproblems can be analyzed at the same time (parallel execution), thereby speeding up the overall analysis process. Decomposition, therefore, improves the efficiency of the design team. Also, the performance of the non-linear optimizers is considerably better for the smaller subproblems than it would be for the entire system[8].

The decomposition method allows the computer codes most often used by the individual disciplinary experts to be retained. Because the subproblems are small and manageable, if the individual codes are incapable of optimization of the subproblem, it is perfectly reasonable to rely on the experience and judgement of the individual designer to come up with an appropriate solution. For example, assume aerodynamics is a subproblem and its individual objective function is maximum lift to drag ratio. If the aerodynamic analysis tool is incapable of changing the design to maximize L/D, the aerodynamics expert could perform the optimization manually based on the inputs given. The results would then be sent to a lower level subsystem in the hierarchical structure. “Human intuition” optimization of this nature is generally prohibitive in the case of classical optimization where the entire analysis process must be quickly repeated a number of times

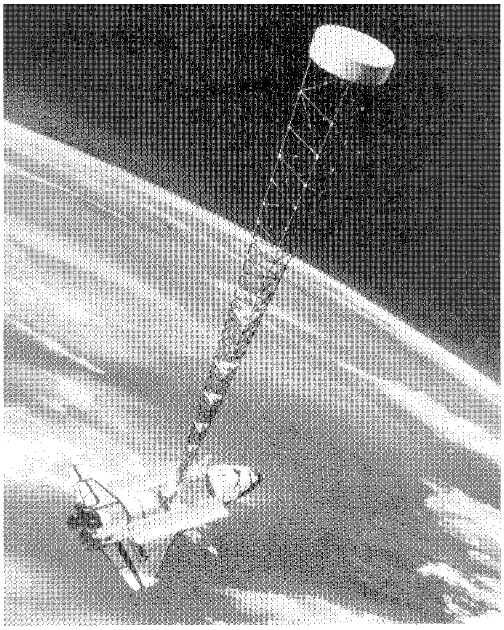

Figure 9 - COFS Experiment

and is, therefore, usually completely automated.

In practice, system decomposition may be best used as a planning and scheduling tool. Especially for new or one-of-a-kind designs, the ability to visualize the subsystems and structure as a top down hierarchy is very beneficial. The process of optimizing the entire system by coordination is a difficult task for most aerospace problems. Additionally, since, like classical optimization, the method varies design variables numerically, it is more difficult to handle discrete variables. Other methods, such as system sensitivity analysis (discussed below) perhaps used in conjunction with decomposition, may be more suitable for the overall optimization process.

Example Application

In reference 9, Padula, et al., apply the method of multilevel decomposition to planning the design of a large, flexible space truss structure deployed from the Space Shuttle. The Control of Flexible Structures (COFS I) experiment used in the example was designed by NASA - Langley to evaluate active controls and system vibration identification techniques to be used in the design of future large space structures (see figure 9 from reference 9). Padula, et al., used system decomposition to divide the problem into smaller subproblems, but they did not document any use of coordination to break system level constraints and objectives into subproblem level equations to be optimized.

The design of large space structures typically involves coupled and often competing design requirements. For example the structure must be lightweight and low volume but robust enough to withstand high launch loads. Once the lightweight structure is put into space, it must be controllable with a minimum of effort if it is to serve some useful purpose as an antenna or a stable platform. Ultimately, a low cost design is also desired.

The COFS experiment consists of a truss structure with sensors placed along its length in order to record the vibration modes of the structural elements. Actuators are also provided to control the structure. The COFS experiment is designed to minimize cost by reducing weight and control power.

From preliminary work, Padula, et al., (Ref. 9) identified 12 separate design variables for the COFS experiment. They include the actuator masses and the number of truss sections. Several intermediate variables (called behavior variables) such as structure bending stiffness and vibrational mode shapes were also identified. The behavior variables are calculated after the design variables have been set. For example, the calculation of the vibrational mode shapes depends on the length of the truss and the location of the sensor and actuator discrete masses. Constraints, such as maximum weight and vibrational frequency targets, were established and written as functions of the design and behavior variables. Finally, the objective function, cost, is evaluated. In the example, cost depends on the system weight and control power requirements.

In reference 10, Rogers takes the initial data relationships defined by Padula, et al., and uses the design tool DeMaid to perform the system decomposition. The 29 design variables, behavior variables, constraints, and the objective function are shown in initial form along the diagonal of the N x N network graph in figure 10 (taken directly from reference 10). The dependencies between the variables are shown as feedforward (above the diagonal) and feedback (below the diagonal) link lines. Feedback implies an iterative loop because data is required before it is calculated. In its initial form, the system contains a number of very large feedback loops and could only be solved as a large and complex optimization problem. The goal of decomposition is to reduce the number, or at least the size, of the feedback loops.

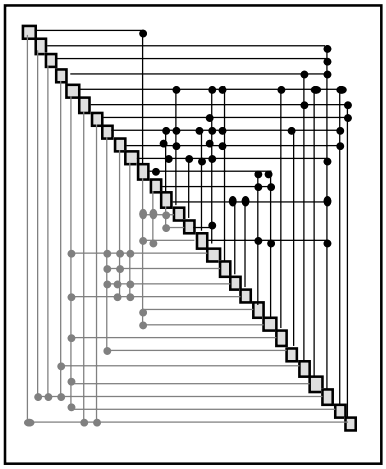

Figure 10 - COFS System Before Decomposition

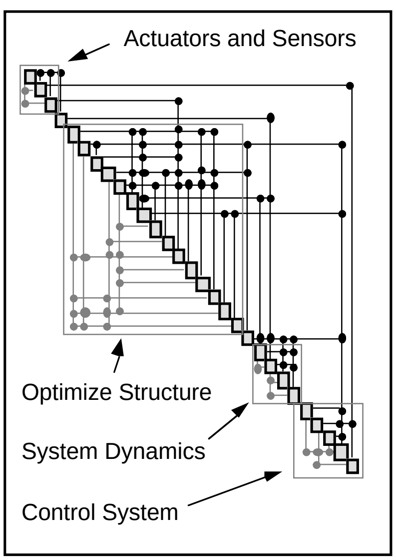

Figure 11 - COFS After Decomposition

After decomposition scheduling, the system is represented by the N by N network shown in figure 11. Notice that the number of feedback loops has been greatly reduced and several distinct analysis circuits (larger groups of modules) have been formed. The individual circuits represent subproblems. The four large circuits created are, starting in the upper left corner: actuator and sensor placement tasks, structural design optimization tasks, system dynamic analysis, and control system design. The structural design task forms the largest circuit, encompassing 15 modules. Notice that the overall analyses should proceed in the logical sequence shown in figure 12 because of the flow of information from the upper level tasks to the lower level tasks. If necessary, iteration on the entire system may be used as indicated in figure 12 (from reference 9).

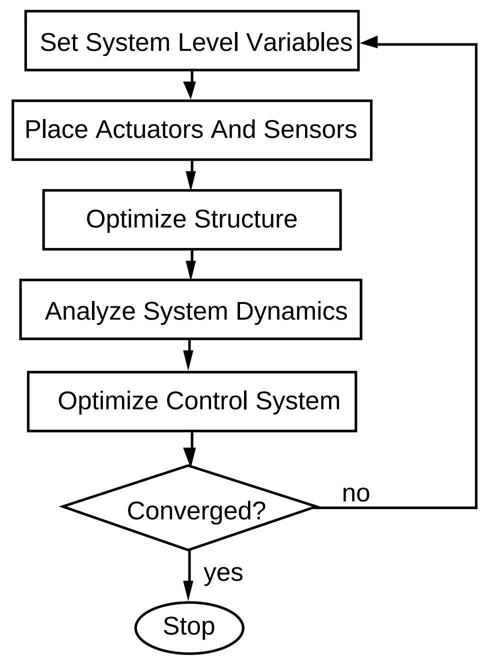

Figure 12- COFS Application Flow Chart

In the COFS example, system decomposition allowed the complex design problem to be reduced to a series of smaller subproblems in the project planning phase. The process also helped organize and visualize the vast number of interactions through the use of the N by N network graph[9]. While the system does require a significant investment of time to identify the design variables and links, it does provide a systematic way to analyze a complex design problem early in the design process[9]. In order to complete the design process, it would be necessary for Padula, et al., to apply coordination or some other technique to the decomposed system and to optimize the design variables and satisfy all the constraints.

Sensitivity Analysis

System sensitivity analysis (SSA) is a multidisciplinary design and optimization method designed to answer "what if" type questions and perform optimization of an entire system. It replaces system level finite differencing (i.e. redesigning the entire vehicle while changing only one variable in order to determine trends and influences on the objective function) with a more efficient, distributed calculation scheme[3]. While classical optimization and system decomposition/coordination are usually limited to smaller sized or less coupled problems, sensitivity analysis is well suited to handle more highly coupled and complex aerospace vehicle design. In fact, it can easily incorporate the techniques of the other two methods. Therefore, system sensitivity analysis may have the widest applicability of any of the numerical techniques discussed in this paper.

System sensitivity analysis treats a system as a coupled set of subproblems - perhaps determined with the help of decomposition. These subproblems are generally traditional design disciplines that retain their existing, more detailed, design codes. By retaining these existing design codes (like NASTRAN or MINIVER), the subproblems can be treated as “black boxes” - exchanging inputs and outputs with other “black boxes” in the overall system. This organization of “black boxes” resembles the organization of some of the multidisciplinary *analysis* tools that have already been developed.

Using a linear matrix equation called the global sensitivity equation (GSE), the SSA method first analyzes the impacts of the various disciplines on each other, then secondly analyzes the impacts of a change in the input design variables on each of the disciplines (while holding other influences constant). For example, an aerodynamics discipline would evaluate its own sensitivities to changes in other disciplines. Increased wing weight might produce a different trim point with a corresponding increase in induced drag. In the second case, assume wing aspect ratio is a design variable. The aerodynamic discipline could calculate the change in trim lift coefficient with increased aspect ratio - temporarily disregarding the fact that an increase in aspect ratio may increase wing weight which indirectly will also affect the trim lift coefficient. Given the necessary sensitivity information, it is possible to solve the GSE for the total influence that changing a design variable will have on the entire system[11]. These System sensitivity derivatives (SSD’s) are total derivatives and are essentially *system* level gradients to be used by a designer either intuitively or numerically to iteratively improve the design. The advantage of the method over top level system finite differencing lies in the computational efficiency, ability to perform subproblem tasks in parallel, and the need to only calculate the disciplinary interdependencies once (per iteration) to analyze all design variable influences. A brief discussion of the mathematical basis of SSA is given below.

A complex coupled system can be thought of as a mathematical function that, for a converged solution, generates a set of output values for a given set of input values. For example, for a given sweep, thickness ratio, aspect ratio, Mach number, etc., a wing will have a given coefficient of lift. If $\mathbf{X}$ is a set of input variables, and $\mathbf{Y}$ is a set of output variables, then:

$$\mathbf{Y} = f(\mathbf{X}) \qquad (9)$$

or written another way:

$$F(\mathbf{X},\mathbf{Y}) = 0 \qquad (10)$$

The output variables are generally used in the evaluation of the objective function or the various constraints. Assume that the parts of the system output vector are generated by distinct disciplines so that the system output vector can be partitioned:

$$\mathbf{Y} = (\mathbf{Y}_1,\mathbf{Y}_2,\mathbf{Y}_3)^T \qquad (11)$$

where, for simplicity, three disciplines have been assumed. For example, aerodynamic coefficients would be generated by an aerodynamics discipline, engine performance would be generated by a propulsion discipline, and the wing stresses would be generated by a structures discipline. Combining eqns. (10) and (11) leads to:

$$F(\mathbf{X},\mathbf{Y}_1,\mathbf{Y}_2,\mathbf{Y}_3) = 0 \qquad (12)$$

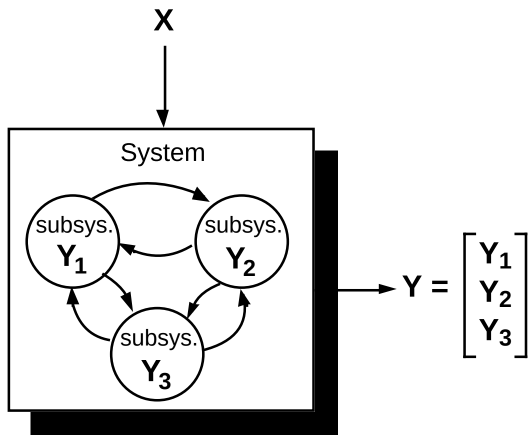

Figure 13 - Coupled System as Three Disciplines

The implicit function theorem allows the equation to be rewritten such that one variable is expressed as a function of the others (assuming decomposition has been performed so that one discipline is not a function of its own outputs[12]).

$$\mathbf{Y}_1 = f_1(\mathbf{Y}_2,\mathbf{Y}_3,\mathbf{X}) = \mathbf{Y}_1(\mathbf{Y}_2,\mathbf{Y}_3,\mathbf{X}) \quad (13a)$$
$$\mathbf{Y}_2 = f_2(\mathbf{Y}_1,\mathbf{Y}_3,\mathbf{X}) = \mathbf{Y}_2(\mathbf{Y}_1,\mathbf{Y}_3,\mathbf{X}) \quad (13b)$$
$$\mathbf{Y}_3 = f_3(\mathbf{Y}_1,\mathbf{Y}_2,\mathbf{X}) = \mathbf{Y}_3(\mathbf{Y}_1,\mathbf{Y}_2,\mathbf{X}) \quad (13c)$$

This system and its corresponding subproblems are shown graphically in figure 13. **X** represents all the inputs, and **Y** is shown in partitioned form.

Taking equation (13a) as a representative example and using the chain rule to write the differential form:

$$d\mathbf{Y}_1 = \frac{\partial \mathbf{Y}_1}{\partial \mathbf{Y}_2}\, d\mathbf{Y}_2 + \frac{\partial \mathbf{Y}_1}{\partial \mathbf{Y}_3}\, d\mathbf{Y}_3 + \frac{\partial \mathbf{Y}_1}{\partial \mathbf{X}}\, d\mathbf{X} \quad (14a)$$

and the total derivative is:

$$\frac{d\mathbf{Y}_1}{d\mathbf{X}} = \frac{\partial \mathbf{Y}_1}{\partial \mathbf{Y}_2}\frac{d\mathbf{Y}_2}{d\mathbf{X}} + \frac{\partial \mathbf{Y}_1}{\partial \mathbf{Y}_3}\frac{d\mathbf{Y}_3}{d\mathbf{X}} + \frac{\partial \mathbf{Y}_1}{\partial \mathbf{X}} \quad (14b)$$

Recall that, due to the coupling of the system, $\mathbf{Y}_1$ is influenced by each of the other subsystems as well as the input variable(s). Equation (14b) states that the total change in the output $\mathbf{Y}_1$ with respect to a change in an input variable is the sum of the changes in each of the other subsystems times their individual effects on $\mathbf{Y}_1$ (these are the partial derivatives) plus the change in $\mathbf{Y}_1$ itself due to a change in the input variable. Performing a similar process on the other two subsystems will lead to the coupled matrix equation known as the Global Sensitivity Equation (GSE).

$$\begin{bmatrix} \mathbf{I} & -\frac{\partial \mathbf{Y}_1}{\partial \mathbf{Y}_2} & -\frac{\partial \mathbf{Y}_1}{\partial \mathbf{Y}_3} \\ -\frac{\partial \mathbf{Y}_2}{\partial \mathbf{Y}_1} & \mathbf{I} & -\frac{\partial \mathbf{Y}_2}{\partial \mathbf{Y}_3} \\ -\frac{\partial \mathbf{Y}_3}{\partial \mathbf{Y}_1} & -\frac{\partial \mathbf{Y}_3}{\partial \mathbf{Y}_2} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{Y}_1}{d\mathbf{X}} \\ \frac{d\mathbf{Y}_2}{d\mathbf{X}} \\ \frac{d\mathbf{Y}_3}{d\mathbf{X}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{Y}_1}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{Y}_2}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{Y}_3}{\partial \mathbf{X}} \end{bmatrix} \quad (15)$$

Note that the partitions of the output vector, $\mathbf{Y}_1$, $\mathbf{Y}_2$, and $\mathbf{Y}_3$ are typically also vectors so the terms of the GSE (e.g. $-\partial\mathbf{Y}_1/\partial\mathbf{Y}_2$) will generally be matrices[12]. The matrix on the left side of the GSE is called the global sensitivity matrix. It contains the sensitivities of each discipline to outputs from other disciplines. The vector on the right side contains the local sensitivity derivatives. It contains the sensitivities of each discipline to changes in the input variables while holding other influences constant. Given values for both of these from discipline level analysis, it is possible to solve the linear problem for the vector of system sensitivity derivatives (left side unknown vector) using existing matrix techniques[13]. By using this process, system level *total* derivatives (SSD's) are calculated from discipline level *partial* derivatives. Note that the vector of local sensitivity derivatives is dependent on the particular input variable (e.g. aspect ratio) being evaluated, and therefore, it must be recalculated for each input variable. However, the global sensitivity matrix is dependent only on the discipline interactions and is only calculated once per iteration.

Using the technique described above, a new set of system sensitivity derivatives (SSD's) is generated for each design variable. These SSD's are essentially gradients of the output variables (weight, cost, etc.), and therefore the objective function, with respect to changes in the design variables (aspect ratio, wing sweep, nose radius, etc.). The calculation of the SSD's is the equivalent of performing a finite difference analysis on the entire system for each design variable. The SSD's can be used intuitively by the designer who would then make changes in the inputs in order to improve the objective function. Alternately, the SSD's could be used in an iterative numerical optimization scheme.

Pros and Cons

System sensitivity analysis (SSA) is very well suited to handle large, highly coupled aerospace vehicle design problems. By making use of decomposition/coordination techniques to break a problem down into a set of smaller subproblems, a complex task can be divided among several design teams. If desired, classical optimization methods can be used to optimize a set of design variables based on the system sensitivity derivatives (SSD's) generated by SSA. Because of this ability to incorporate and expand on previously discussed techniques, SSA may have the widest applicability of the numerical techniques discussed here.

Once a system is divided into subproblems, they can be treated as "black boxes" providing outputs to and receiving inputs from other "black boxes". This capability is of particular interest because it allows certain disciplines to retain their existing detailed deign tools (e.g. NASTRAN) to process inputs and create necessary outputs. Modification of existing codes is usually not necessary - except to perhaps speed data exchange.

SSA, by nature of its distributed network of subproblems, allows the parallel execution of some of the subproblem tasks. For instance, local aerodynamic sensitivities to wing sweep and local structural sensitivities to wing sweep could be performed simultaneously by separate design groups and later combined to form the local sensitivity derivative vector. By taking advantage of parallel execution of design tasks, the iteration time and the overall design time can both be shortened.

On the negative side, a highly non-linear design may necessitate the frequent reevaluation of the global sensitivity matrix. One advantage of the SSA method lies in its ability to save computational time by using the same global sensitivity matrix for several iterations of design variable changes. Non-linear problems may erode some of these time savings.

Like all numerical optimization schemes, SSA prefers to deal with smooth, continuous functions in order to evaluate derivatives. In typical aerospace vehicle design, discrete variables are highly likely to be present. For example, the number of engines or structural material type. Techniques exist within numerical optimization to deal with discrete variables, but the methods work much better with continuous variables.

Finally, the system sensitivity analysis method is highly numerically intensive during the evaluation of the global and local sensitivities. In practice, the method may be difficult to apply to some disciplines that are not used to working with sensitivities to given inputs (e.g. cost sensitivity to wing sweep).

Example Application

In reference 14, Barnum, et al., students at Virginia Polytechnic Institute and State University, apply the method of system sensitivity analysis to the optimization of a three surface civil transport aircraft. The project was part of a senior design course, and it started with a feasible baseline aircraft design from a traditional trial and error design sequence. The goal of the project was to use SSA to provide additional direction to minimize the aircraft take-off gross weight (TOGW). Among the constraints were the aircraft range, number of passengers, stability requirements, take-off distance (balanced field length, BFL), and maximum wingspan. The baseline design is shown in figure 14 taken directly from reference 14.

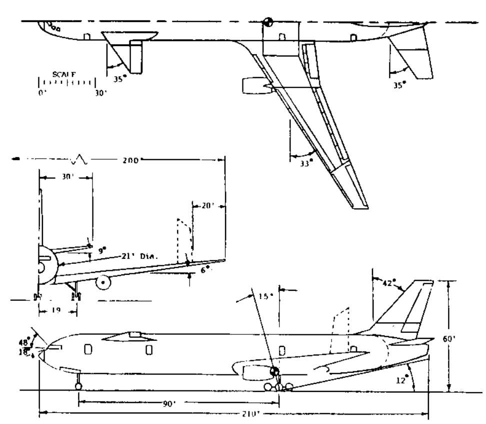

Figure 14 - Baseline Aircraft (from reference 14)

During the process of setting the problem up as a system sensitivity analysis problem, several decomposition variations were attempted in order to best represent the interactions between subproblems. The final decomposition made use of six different subproblems (disciplines). Weight calculation, cruise performance, and takeoff performance were treated as separate parts of the ACSYNT aircraft synthesis tool.

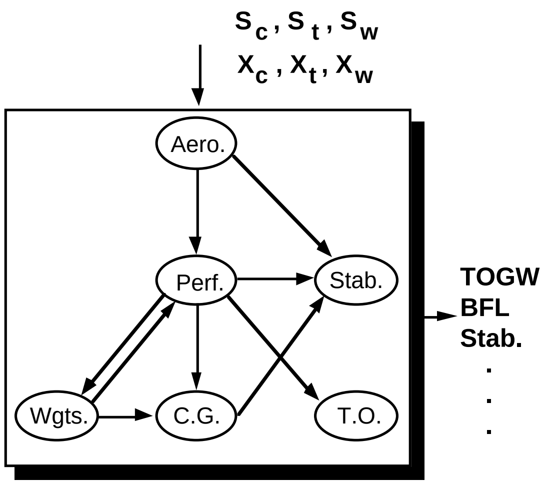

Figure 15 - Subproblem Set (from reference 14)

Aerodynamics were calculated by the VLM vortex lattice method code and FRICTION, a code to calculate viscous drag. Digital Datcom was used to evaluate the stability subproblem. Reference 14 contains bibliographic citations for each of the codes used. The final set of subproblems and links is shown in figure 15 taken from reference 14.

The input design variables chosen for the system were the planform areas of each of the three surfaces (canard, wing, and tail), S_c, S_w, S_t, and the locations of the three surfaces measured from the nose of the aircraft, X_c, X_w, X_t. Using these disciplines and design variables, the global sensitivity matrix and local sensitivity vectors (1 for each of the 6 design variables) were calculated for the baseline aircraft. The initial set of system sensitivity derivatives (SSD's) were calculated, and a linear programming optimizer used this data to make appropriate changes in the design variables. The first iteration resulted in a 1.8% decrease in the aircraft TOGW. Three more iterations were then completed according to the flow chart shown in figure 16 which is based on figure 3 from reference 14.

For each step, outputs like weights and take-off distance were calculated by linear extrapolation from the previous values using the SSD's and the design variable deltas. That is, the aircraft was not completely reanalyzed between each iteration. Rather, a first-order Taylor series approximation was used to predict the necessary output variables. A move limit of ±10% made the linear extrapolation more valid. After each iteration, however, a new global sensitivity matrix and set of local sensitivity vectors were calculated based on the extrapolated values. Because the global sensitivity matrix was not heavily populated, it was possible to recalculate the entire matrix between each iteration. Normally, this is done only for very non-linear problems.

After four iterations, the solution began to show signs of convergence. Indications were that further iterations would soon locate a converged minimum TOGW. In the interest of time, however, the optimization was stopped at this point. The series of linear approximations predicted a TOGW savings of 4.94% less than the baseline design. A final complete analysis of the aircraft with the design variables set a their "improved" values showed an actual TOGW savings of 2.57%. The numbers were slightly different because of the non-linearities in the analysis, but overall, the weight savings was significant considering the design started with a "good" aircraft. All of the constraints were satisfied by the final aircraft. Each of the surface area design variables, S_c, S_w, and S_t, was decreased from the baseline to the final aircraft, and each of the surfaces was moved toward the nose of the aircraft. Figure 17, taken directly from reference 14, shows a planform view of the aircraft before and after the optimization process.

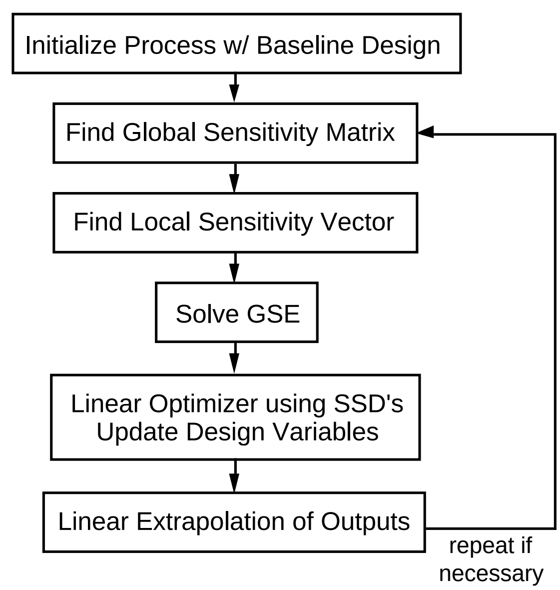

Figure 16 - SSA Method Flowchart (from ref. 14)

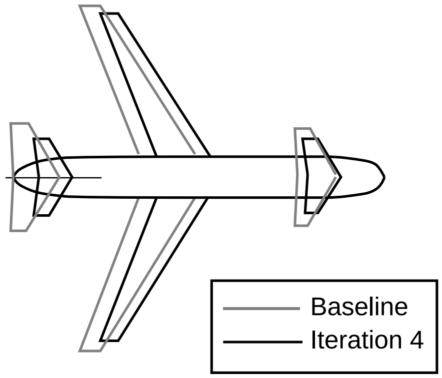

Figure 17 - Changes to Baseline Design (from ref. 4)

Table 1, based on data from reference 14, shows the history of selected design variables, the TOGW, and the BFL for the baseline design and the four iterations.

Table 1 - Iteration Results for 3 Surface Aircraft

Iter.	X_c (ft)	S_c (ft^2)	TOGW (lb)	BFL(ft)
Base	29.00	837.00	521,121	6400
1	26.13	753.30	511,671	9690
2	26.13	677.87	504,639	9930
3	26.13	610.17	498,205	10,000
4	26.68	604.74	495,365	10,000

It is interesting to note that the time to complete the first iteration was 12 days. The final iteration took only 3 days as the design team became more experienced with the method. While the very rapid turnaround time of the simpler, classical optimization method is never achieved, this example of the use of SSA certainly demonstrates the utility of the method when a higher level of analysis detail is required. In addition, the method was able to produce an improvement in the objective function even for a design that had already undergone a "traditional" design process and was considered a good design.

Taguchi Methods

Taguchi methods are essentially "smart" parametric design methods that allow a designer to explore a given parametric space with a minimum number of point design assessments. Unlike the numerical methods discussed above, Taguchi methods do not depend on the calculation of a numerical derivative or gradient as part of an iterative design process. Rather, the entire design space is explored through a series of single point designs that have been strategically and mathematically selected in order to determine all of the effects of and several of the interactions between the input variables. This data is then used to predict the optimum combination of the design variables that will minimize the objective function and satisfy all of the constraints. In addition to locating the minimum of the objective function, the Taguchi method provides information on variable trends and noise sensitivities thereby enabling a robust, tolerant design, as well as an optimum one, to be selected. The method is easy to apply, and it essentially transforms a complicated, iterative design process into a series of applications of the multidisciplinary analysis tools that have already been developed for aerospace vehicles (AVID, EASIE, etc.). It is also well suited to the use of discrete variables. While no such background was provided for the previous methods reviewed, a brief history of the development of Taguchi methods may be of interest here.

Taguchi methods (a specialized application of statistical methods called experimental design or design of experiments methods) are named for the Japanese engineer, Genichi Taguchi, who refined and simplified the existing methods of experimental design through the use of orthogonal arrays[15]. The statistical methods known as design of experiments were originally formalized by the British statistician R. A. Fisher in the 1920's[16]. Using arrays called partial-factorial designs, Fisher showed that the experiments run for all of the combinations and levels of a given set of design parameters (a full factorial array) could be reduced to a more manageable, but still statistically meaningful subset. His methods were applied to agricultural yields[16]. While trying to improve the off-line quality control of the Japanese communications system after World War II, Taguchi struggled with the same problem of very large combinations of parameters[17]. Taguchi realized that the quality of a given product must be "designed in" during the *early* stages of the overall design process. If the product was designed properly off-line (i.e. before it actually went into production), it would be fairly insensitive to the uncontrollable noises it may encounter during the manufacturing process and therefore fewer defects would be produced and money could be saved. Given that there were potentially thousands of design variables to be checked, testing all of the combinations to produce the most robust product was a huge problem. The existing statistical methods embodied by Fisher's design of experiments theory were available, but they were generally thought to be too complicated or unwieldy for the average engineer to use. Taguchi used simple orthogonal arrays to reduce the complexity and the number of experimental runs involved in the classical design of experiments method, and, although his method sacrificed some of the parametric interactions, it still remained statistically valid[18]. Taguchi published the orthogonal arrays used by his method, and he essentially "cook-booked" the analysis techniques that allow the designer to efficiently analyze the experimental results in order to determine the most important parameters. Since the mid-1960's all Japanese engineers have been trained in the use of Taguchi methods[19]. In the early 1980's, the Taguchi method began to be used by engineers in the United States - primarily in the automotive and electronics industries[19]. The method does appear to have applicability to the early stages of aerospace vehicle design. A brief mathematical discussion begins below.

The original design of experiments theory was based upon techniques to reduce the number of experimental runs necessary to characterize a design space. Faced with the large number of experiments required to analyze *all* of the combinations of a set of variables, R. A. Fisher developed a way to represent the full factorial array of experiments by smaller, but still statistically meaningful arrays called partial factorial designs[18]. The partial factorial arrays strategically chose points (combinations) from the full factorial array in order to provide necessary trends and interactions. The primary concern is that the new set of experiments remain "unbiased" with respect to all of the variables. By unbiased, we mean that the results are not more heavily influenced by one parameter setting than another. Biased solutions typically result from the one-at-a-time optimization performed by typical parametric studies.

Consider a simplified model of the system being designed. The objective function, $\mathbf{Y}$, is a function of the settings of the design variables, $\mathbf{X}$.

$$\mathbf{Y} = f(\mathbf{X}) \tag{16}$$

Then, assuming k design variables, a first order approximation by Taylor series expansion would yield the following formula that enables the effect of each parameter to be determined.:

$$\mathbf{Y}_2 = \mathbf{Y}_1 + \sum_{i=1}^{k} \frac{\Delta \mathbf{Y}}{\Delta x_i} \Delta x_i \tag{17}$$

As an aside, the first order model is usually inadequate for most designs. It is used in this case for illustrative reasons. A second order model will nominally enable the capture of curvature and interaction effects (i.e. non-linear effects) with a full factorial array. 3 level Taguchi arrays can capture some of these effects. See reference 18 for more details.

Returning to the first order example in equation (17), there are k unknown sensitivity coefficients. In addition, $\mathbf{Y}_1$ may be an unknown. A full-factorial array that runs all of the combinations of the variables would produce an unbiased solution. Taguchi's method (using orthogonal arrays) can reduce the number of required experiments significantly. Figure 18 illustrates the full-factorial and orthogonal arrays graphically for a sample case of 3 parameters at 2 levels each. The experimental combinations are marked by circles. A full-factorial would use 8 experimental runs (2^3) representing the 8 corners of the cube in figure 18a. An unbiased, average $\Delta\mathbf{Y}/\Delta x_1$ would then be calculated:

$$\frac{\Delta \mathbf{Y}}{\Delta x_1} = \frac{1}{4\Delta x_1}\left[(y_4-y_1)+(y_3-y_2)+(y_7-y_6)+(y_8-y_5)\right] \tag{18}$$

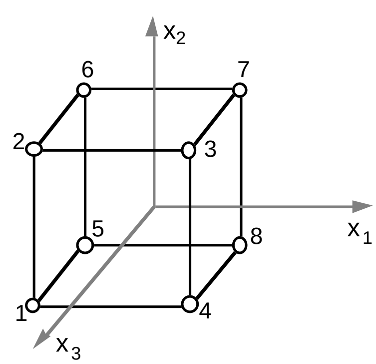

Figure 18a - Full Factorial Array

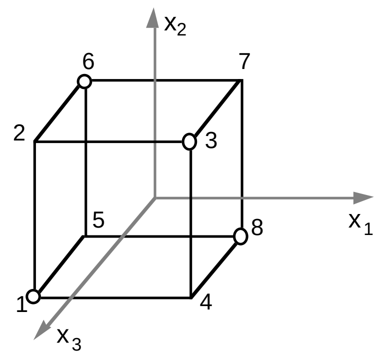

Figure 18b - L_4 Orthogonal Array

The orthogonal array (the L_4 in this case) is represented by figure 18b. It is also unbiased because for each level of a parameter, the levels of all of the other parameters are represented equally. So, in only 4 runs, an unbiased, average $\Delta\mathbf{Y}/\Delta x_1$ would be calculated:

$$\frac{\Delta\mathbf{Y}}{\Delta x_1} = \frac{1}{\Delta x_1}\left[\frac{(y_3+y_8)}{2} - \frac{(y_1+y_6)}{2}\right] \qquad (19)$$

For comparison, figure 18c shows an array that would produce biased results (slanted toward lower x_1's). The orthogonal array is the minimum, and therefore most efficient, unbiased set of experiments that can still capture all of the main effects of a system. However, the ability to capture *all* of the interactions is lost when going from full factorial arrays to orthogonal arrays[18]. In practice, this isn't a significant issue because we are primarily concerned with the main effects and only a few interactions.

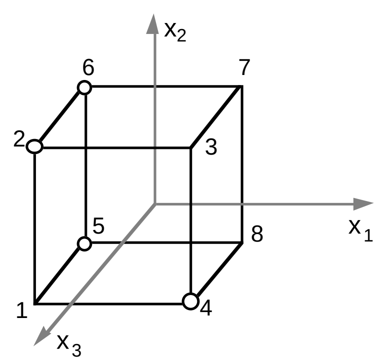

Figure 18c - Biased Array

For experiments with more levels and parameters, it becomes necessary to describe the orthogonal arrays mathematically rather than graphically. As mentioned above, orthogonal arrays provide a method for reducing the number of experimental runs required to characterize a design. The most efficient set of experimental runs is the number that is exactly equal to the number of degrees of freedom of the design. Assume we have four parameters (A, B, C, and D) that we want to check at two levels. For example, parameter A may be engine thrust, and we want to check the influences of a high and low thrust setting. In addition, we want to check the interactions between A&B, A&C, and B&C. The calculation of the mean of all the responses represents one degree of freedom. There is also a degree of freedom associated with changing each main variable from the base level to its other level. In addition. each 2 level interaction has a number of degrees of freedom equal to the product of the number of changes that each of the interacting variables is allowed to make. In this case, we can mathematically calculate the degrees of freedom (DOF) as:

$$DOF = M + V * (L-1) + I * (L-1)^2 \qquad (20a)$$

where,

- I = number of interactions being analyzed
- L = number of levels of the main variables
- M = 1 (representing the overall mean)
- V = number of main variables

so, for our example

$$DOF = 1 + (4) * (2-1) + (3) * (2-1)^2 = 8 \qquad (20b)$$

Therefore, for the highest efficiency, 8 experiments should be run to capture all of the primary data. For comparison, a full exploration of the design space with all combinations of the parameters would require 2^4 or 16 runs. As the number of variables and number of levels increase, the relative advantage of the Taguchi method also increases. For example, we could explore the main effects of 13 variables at 3 levels each with only 27 experiments using Taguchi's L_{27} array. The full-factorial array would require 3^{13} or 1,594,323 experiments! Taguchi has published standard orthogonal arrays for various required number of experiments and levels of parameters. For our example, Taguchi's L_8 array is the standard array to be used. Table 2 shows the L_8 array.

Table 2 - Taguchi L_8 orthogonal array

Exp	A	B	A&B	C	A&C	B&C	D
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

Here, our interactions have been placed in columns 3, 5 and 6 (see reference 16 for more information). In the array, the 1's represent the lowest level of the parameter (i.e. lowest thrust) and the 2's represent the highest level. The 8 required runs are read horizontally across each row with the four parameters being set accordingly. The interactions are calculated after all 8 runs have been performed.

The orthogonality of the array is vital to ensuring that the experiment can be done in a minimum number of experimental runs. An orthogonal array is one in which the columns of its *contrast* are all orthogonal or mutually independent vectors[16]. Since they are orthogonal, no column can be created by a combination of any of the other columns. Therefore, unnecessary repetition is avoided during the analysis. The *contrast* of an array is an array that replaces the levels of an original array (1 and 2) with weighted values, w_i. The weighted values are constrained to add up to zero[16]. For example, the most typical way to create a contrast of a two level Taguchi array is to replace all of the level 1's with -1 and all of the level 2's with +1 weights. The sum of the weights is -1+1 = 0. The contrast for our L_8 array is shown in Table 3.

Table 3 - Contrast of Taguchi L_8 array

Exp	A	B	A&B	C	A&C	B&C	D
1	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	+1	+1	+1	+1
3	-1	+1	+1	-1	-1	+1	+1
4	-1	+1	+1	+1	+1	-1	-1
5	+1	-1	+1	-1	+1	-1	+1
6	+1	-1	+1	+1	-1	+1	-1
7	+1	+1	-1	-1	+1	+1	-1
8	+1	+1	-1	+1	-1	-1	+1

We can verify that the first two columns of the contrast array are orthogonal by evaluating their dot product.

$$(\text{col } 1)\bullet(\text{col } 2) = 1+1-1-1-1-1+1+1 = 0 \qquad (21)$$

All other columns in the contrast are similarly orthogonal. The contrast can also be calculated for 3 or more level Taguchi arrays.

Another characteristic of orthogonal arrays was demonstrated in our previous graphical example and is called the balancing property[16]. That is, for every set of 2 columns, the pairs of levels 1 and 2 occur in all combinations, and they occur an equal number of times. For the L_8 array, the pairs of levels (1,1), (1,2), (2,1), and (2,2) each occur twice for any 2 columns. As a result, the effects of changing parameter levels is distributed evenly between any two columns. The balancing property is a sufficient condition to prove the orthogonality of an array[16].

Taguchi uses a L with a subscript to reference the standard arrays used by his method (e.g. L_8, L_9, L_{18}, L_{27}, ...) where the subscript indicates the number of experiments to be calculated. Once the experimental parameters, ranges, and levels have been identified, an appropriate array is chosen from the set of standard arrays. It should be noted that not all experiments fit nicely into a standard array. In those cases, either the array or the parameters are modified in order to fit a standard array.

Now that the background for the orthogonal array has been established, we can begin to discuss the application of the method. The Taguchi method can be divided into two primary design tasks. Analysis of the mean (ANOM), or response table analysis, involves determining the trends of the variables and their interactions over a specified range in order to optimize the problem. Signal-to-noise ratio analysis involves determining the affects of noises or uncontrollable variations on the design in order to formulate the most robust design.

The ANOM technique makes of use the orthogonal arrays discussed above to minimize the number of experimental design points necessary to characterize the entire problem. The value of the objective function is calculated for each row of the appropriate Taguchi array using a multidisciplinary analysis tool. For our example using the L_8 array, assume that minimum gross weight is the objective function. Then the first experiment would calculate gross weight with the input parameters A, B, C, and D all being set at their low (or 1) values. The columns representing interactions between parameters (e.g. A&B) are ignored during the process of running the experiments/analysis. The interactions are later calculated from the objective function results using the levels listed in the columns of the Taguchi array. The analysis proceeds until all 8 experiments have been performed and the 8 resulting gross weights have been recorded.

Note that, to this point, this process is more of an analysis process than a design process. There is no attempt to feedback any changes or improvements to the design parameters. The inputs are considered fixed by the method, and the analysis is then carried out using a multidisciplinary analysis tool or series of tools. The method could easily make use of existing detailed design codes, treating them as "black boxes". Iteration between these existing design codes may still be necessary (for example, iteration between performance and weights) in order to calculate the objective function for a given experiment. The Taguchi method is fundamentally different from the numerical optimization schemes which seek to feedback changes in the design variables based on the gradient of the objective function. These schemes change the design variables between each iteration in order to step toward a progressively better solution. The Taguchi method, however, explores the entire design space with a coarse grid and later uses all of the information to determine the optimum combination of variables.

Once all of the experiments have been run, the trends associated with changes in each parameter are determined from analysis of the mean (ANOM). The arithmetic average or mean of all of the objectives is calculated for all runs with a parameter set at level 1. Then the mean is found for all experiments with the parameter set at level 2. These two means are either recorded in a mean response table or plotted in a linear graph vs. the parameter levels. The graph shows the *overall* trend of choosing one level over another. For our L_8 example, recall that parameter A in column 1 is engine thrust. Out of the 8 experiments performed using the L_8 array, there are 4 with engine thrust at its low value and 4 with engine thrust at its high value. The averages of these two set of runs will show the overall trend of changing engine thrust even though

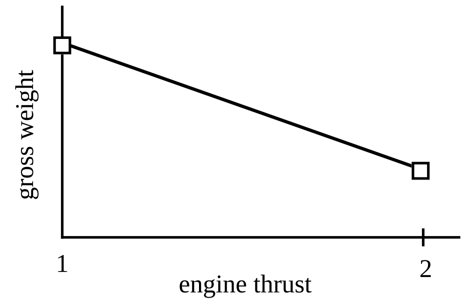

Figure 19 - ANOM result of 2 level Taguchi Array

the other parameters are changing. If the results show that the higher thrust tends to produce a lower gross weight vehicle on average, then we would chose the higher thrust for the optimum solution (see figure 19). A similar process is performed for each of the main parameters.

Interactions may influence the choice of the levels of each parameter. That is, the trend associated with one variable may strongly depend on the level of another variable. In that case, it is said that an interaction exists between the two parameters and the parameters are optimized as a pair. Taguchi recommends a confirmation run using the final set of optimum parameters in order to verify the optimum and to avoid unidentified interactions or non-linearities[16]. In order to refine the parameter ranges, a designer may chose to iterate several times using a Taguchi approach. Each successive iteration may or may not use the same Taguchi array as new interactions are found or main parameter ranges have been sufficiently narrowed down.

An extension of the ANOM technique is the more powerful technique for determining the trends and interactions of the main variables called Analysis of the Variance (ANOVA). ANOVA, while not used by many designers and not necessary for all situations, uses additional statistical techniques to further analyze the problem and provide additional data. For example, ANOVA uses the standard deviation of the mean results from a particular parameter to insure that the changes in the overall response are statistically valid[17]. That is, are the results inside or outside of the statistical noise of the experiments? In addition, simple averaging of the responses as performed by ANOM cannot provide interaction data for all of the variables in question. ANOVA can be employed to provide some additional data. Reference 20 (Pilon) contains ANOVA data from an experimental application of the Taguchi method to the design of a plastic container using finite element analysis. Using ANOVA, the designer was able to determine relative importance of the various design parameters on the overall product design[20]. References 16 and 17 provide additional information about the analysis of variance technique.

The second primary analysis tool within the Taguchi method is signal-to-noise ratio (S/N) analysis. As mentioned earlier, a good design is one that is fairly insensitive to uncontrollable outside influences. For example, a launch vehicle that is more tolerant to unexpected weight growth is more desirable than one that is not. Once the noise factors and appropriate levels are identified, a second orthogonal array is selected from Taguchi's list to create a noise array (also called the "outer array")[17]. The noise array is used in conjunction with the original controllable factors array (or "inner" array) such that for each row of the inner array, experiments are performed for all of the rows of the outer array. If the inner and outer arrays are both L_8 arrays, then the result would be 64 evaluations of the objective function. Using the objective function data for each case, an appropriate signal-to-noise ratio is calculated that, in effect, represents the ratio of the effect of the parameter on the mean of the objective function to the sensitivity of that parameter to the uncontrollable noises. A higher signal-to-noise ratio is the most desirable because it indicates a parameter that controls the objective function without being overly sensitive to uncontrollable noises. For our L_8 controllable factors array, assume we want to test the sensitivity of the system to 3 noise factors at two levels each (e.g. 10% and 15% weight growth, 2 levels of cross winds during landing, and 2 launch delays). We can construct the noise array using a L_4 Taguchi orthogonal array. So, for each of the 8 rows of the original array, we now perform experiments at 4 different noise combinations. The result is 32 evaluations of the objective function - vehicle gross weight (see figure 20)

Since we are trying to minimize the vehicle weight, the appropriate signal-to-noise ratio to use is the "smaller-the-best" S/N. The S/N for each row of the controllable factors array is calculated using the following equation from reference 16.

$$S/N = -10*\log_{10}\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right) \qquad (22)$$

where,

n = number of rows in the noise array (4)
y_i = objective function at each noise column

After a single S/N is calculated for each row, an average S/N is then calculated for each of the controllable design parameters at each of its settings (similar to the ANOM calculation). Higher S/N's indicate a statistically lower vehicle weight even when noise is included and are, therefore, better settings for

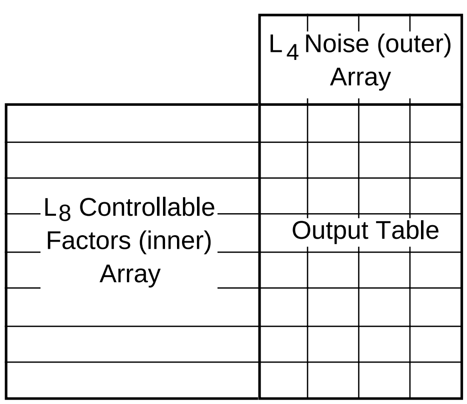

Figure 20 - Inner and Outer Arrays

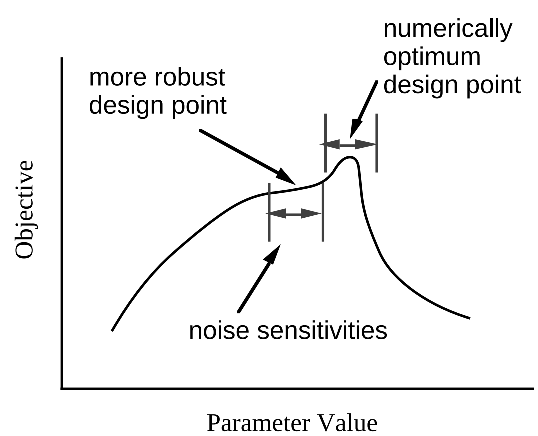

Figure 21 - Robust Design vs. Numerical Optimum

the parameters. Using S/N information, a designer can identify which parameter levels are most sensitive to uncontrollable noises. Therefore, a more robust, noise tolerant system can be designed. Figure 21 shows a case where knowledge of the noise sensitivities allows a designer to select a more robust design point rather than risk a more “optimum” setting that is overly sensitive to uncontrollable factors. The S/N formulas for cases where a specific output value is being targeted (i.e. nominal-the-best S/N) include more terms such as the standard deviation and mean of the row[16]. In all S/N cases, however, a larger S/N is a more desirable case.

While many applications of the Taguchi method do not make use of signal-to-noise ratio techniques, it is perhaps the greatest strength of the method. In reference 21, Byrne and S. Taguchi give an example of signal-to-noise ratio analysis applied to the design of an elastomeric hose connector where the controllable factors are adhesive concentration, connector wall thickness, insertion depth, and interference fit and the noise factors are conditioning time, conditioning temperature, and conditioning humidity[21].

Pros and Cons

The Taguchi method is very easy to apply and does not require numerical gradients and derivatives to be generated for each step in an iteration process. The experimental runs, with or without noise factors, to be analyzed are established from the beginning of the design process. Existing detailed analysis codes can be retained as “black boxes”. The method does not require the analysis experts to provide any “new” information as part of their individual analysis processes. For that reason, Taguchi methods may be easier to adopt by an established design team which may tend to resist the “cultural change” associated with the sensitivity coefficients or partial derivatives required by most numerical optimization schemes.

Because parametric ranges and levels are used, the process lends itself very well to the use of discrete variables. Structural material type, for example, could be one of the input parameters with the 2 levels representing two completely different materials. Numerical optimization techniques would have a very difficult time dealing with such a parameter because derivatives do not exist for discrete variables.

In addition, the Taguchi method tends to characterize the entire design space rather than just finding the optimum answer. Interactions, parametric trends, and noise variances are all identified by the method. Armed with such information, a designer may have more confidence in the final design.

On the negative side, the results from the Taguchi method are not truly optimums in the sense of several decimal place accuracy. The results will only show trends over the range and levels given by the designer. If the initial ranges are incorrect, the method will indicate that the ranges should be refined and the method should be repeated. Even if the initial range does enclose a minimum, the grid may be too coarse to identify an optimum. Repeated iteration may be necessary to narrow a solution down to an accurate optimum. On the other hand, the method is designed to put less emphasis on the most optimum solution and more on the most robust solution. So 2 or 3 iterations may be all that is required in order to find a robust, if not exactly optimum, solution.

Because of the Taguchi method’s ease of use, ability to deal with discrete variables, and ability to find a near optimum , if not an exact optimum, it may be most applicable to the early phases of a vehicle design - where configuration and material options still remain undecided. Using the method, variable trends and interactions can be identified to enable one or two designs to be selected for more detailed study.

Example Application

In reference 22, Stanley, Unal, and Joyner apply the Taguchi method to the design of a dual mixture ratio single stage to orbit (SSTO) rocket powered launch vehicle. The design task was to determine the optimum values of several propulsion parameters associated with the dual mixture ratio engine, identify key interactions, and to compare the results to a single mixture ratio SSTO vehicle also optimized using the Taguchi method. In both cases, the engine made use of a 2 position nozzle in order to maximize expansion efficiency. The objective for both vehicles was to minimize dry weight for a fixed payload and polar orbit destination. Only the ANOM technique was used in this example. Figure 22 shows the SSTO vehicle being optimized.

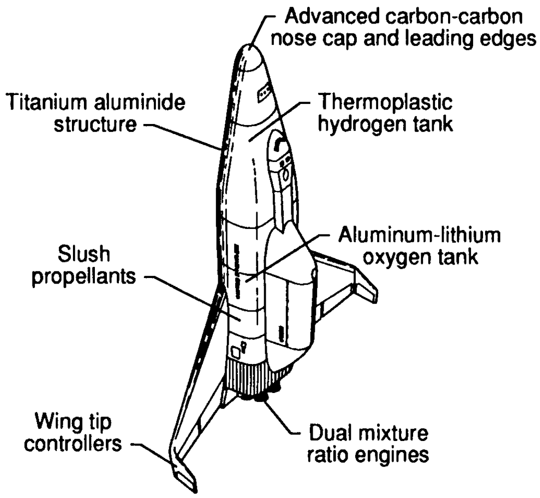

Figure 22 - SSTO Dual Mixture Ratio Launch Vehicle

The computational analysis tools used to analyze both the dual mixture ratio (MR) and the single MR vehicles were the Program to Optimize Simulated Trajectories (POST), CONSIZ (a configuration weights and sizing code), and an engine performance tool. The variables used to optimize the dual mixture ratio engine were initial engine chamber pressure P_{c1}, the mixture ratios MR_1 and MR_2 (liquid oxygen to liquid hydrogen by weight), the 2 engine area ratios for the 2 position nozzle AR_1 and AR_2, the initial thrust to weight of the vehicle T/W_o, and the Mach number at which the mixture ratio is transitioned, M_{tr}. The engine parameters for the single mixture ratio engine were P_c, MR, AR_1, AR_2, and T/W_o.

Previous work has shown that a dual mixture ratio engine can save vehicle dry weight by reducing the volume of LH2 propellant used and thus allowing lower hydrogen tank weights. This is accomplished by increasing the mixture ratio, LOX/LH2, of the engine to above 10 early in the launch trajectory. The high mixture ratio produces a high thrust (although lower efficiency or specific impulse) due to the higher mass flow rate and high propellant bulk density. Later in the trajectory, the mixture ratio is reduced to a more typical value around 7 for a better I_{sp}. The result is that more oxygen, but less hydrogen is burned for the entire trajectory compared to a nominal mission. Because of its comparatively lower density, the hydrogen savings translates to a significant tank weight reduction. The tank weight savings equates to a smaller and cheaper vehicle.

In order to perform the analysis, Stanley, et. al., selected the L_{27} Taguchi array which allows for the use of 7 main variables at three levels each and 3 interactions. The ranges of the optimization parameters chosen for the dual mixture ratio SSTO vehicle are shown in Table 4 (from reference 22).

Table 4 - Input Parameters and Levels

Inputs	Levels		
	L	M	H
AR_1	20	40	60
AR_2	60	110	160
MR_1	10	12	14
MR_2	5	6	7
P_{c1}	3,000	3,850	4,700
M_{tr}	1.5	3	4.5
T/W_o	1.2	1.35	1.5

The analysis proceeded according to the 27 experiments outlined by the L_{27} array using the three analysis tools described earlier. For each of the 27 experiments, the dry weight was found and recorded. Preliminary analysis of the results indicated that a strong interaction may have been neglected. Therefore, the interactions were all reevaluated and the process was repeated using an L_{18} array. The analysis process followed the flow chart shown in figure 23.

Using analysis of the mean (mean response table) techniques, the optimum settings for each variable were determined. Also, a strong interaction

Table 5 - Mean Dry Weights Response Table

Levels	Parameters: MR2	Mtr	AR1	Pc1	T/Wo	MR1	AR2
L	151,731	138,410	143,714	142,996	135,654	133,280	142,183
M	N/A	137,636	136,652	140,549	139,218	138,477	138,321
H	126,965	142,000	137,679	134,500	143,174	146,288	137,541

between the transition Mach number, M_{tr}, and the second mixture ratio, MR_2, was identified. Table 5 (taken from reference 22) shows the response table of the ANOM analysis for the subsequent L_{18} array. Each entry in the table represents the average of all runs where the given parameter was set at the appropriate level. The numbers in the table are all dry weights in pounds. The standard procedure would be to select the setting for each variable that produces the lowest mean dry weight. For example, the lowest mean dry weight in the P_{c1} column, 134,500 lbs., is produced when P_{c1} is set at its highest setting, 4700 psia. The results obtained this way would be:

MR_2 = **H,** M_{tr} = **M**, AR_1 = **M**, P_{c1} = **H**,
T/W_o = **L**, MR_1 = **L**, AR_2 = **H**

However, in this case, a strong interaction exists between M_{tr} and MR_2. Table 6 (taken from reference 22) shows that it will be most beneficial to set MR_2 at its highest value and M_{tr} at its lowest value.

Table 6 - Mtr&MR2 Interaction

	Mtr = L	Mtr = M	Mtr = H
MR2 = L	155,527	148,493	151,173
MR2 = H	121,293	126,778	132,826

The main effect response table has already identified MR_2 as being optimum at its highest level. However, M_{tr} should be changed from **M** to **L**. Therefore, the final optimum chosen by Stanley, et al., is:

MR_2 = **H,** M_{tr} = **L**, AR_1 = **M**, P_{c1} = **H**,
T/W_o = **L**, MR_1 = **L**, AR_2 = **H**

or, in terms of actual values:

MR_2 = **7**, M_{tr} = **1.5**, AR_1 = **40**, P_{c1} = **4,700** psia,
T/W_o = **1.2**, MR_1 = **10**, AR_2 = **160**

The verification run using the parameters set at these levels produced an optimum dry weight for the dual mixture ratio vehicle of 109,400 lbs. This optimum is less than any of the individual experimental runs performed during the analysis - lending credibility to the method used. As a final result, Stanley, et al., showed that the single mixture ratio SSTO launch vehicle designed using the Taguchi method had a minimum dry weight of 117, 000 lbs indicating an advantage for the dual mixture ratio vehicle[22]. It is likely that the optimum value for dry weight for both vehicles could have been improved somewhat by refining (i.e. narrowing the grid) the parametric ranges and repeating the analysis.

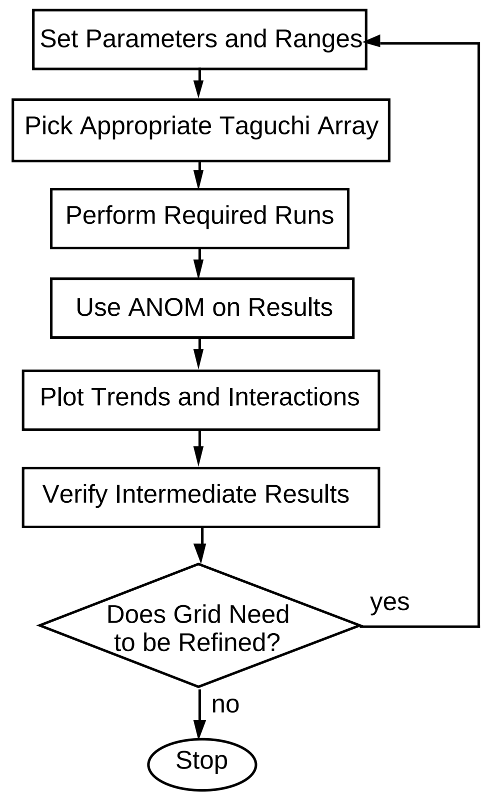

Figure 23 - Taguchi Method Flow Chart

Summary

In summary, four methods for preliminary aerospace vehicle design have been reviewed in this paper. The first three methods (classical optimization, system decomposition, and system sensitivity analysis) make use of numerical optimization techniques and numerical gradients to feed back changes in the design variables. The optimum solution is determined by stepping through a series of designs toward a final solution. Of these three, system sensitivity analysis (SSA) is perhaps the most applicable to large scale, highly coupled vehicle design where an accurate minimum of an objective function is required - perhaps to a design that has moved past the early conceptual design phase into the preliminary design phase. It allows designers to retain their existing design tools for more detailed analysis. Using SSA, several tasks can be performed in parallel - thereby speeding up the design process. Finally, the techniques of classical optimization and decomposition can be included in SSA resulting in a very powerful design method. For example, if decomposition is used to produce 2 subproblems that are still internally highly coupled, SSA can be used on each subproblem separately, and the results can be recombined using coordination.

The Taguchi method, unlike the other methods reviewed, is more of a "smart" parametric design method that analyzes variable trends and interactions over designer specified ranges with a minimum of experimental analysis runs. It also includes a technique for determining the sensitivities to uncontrollable noise factors enabling the selection of a more robust design. The strengths of the Taguchi method are its relative ease of use, ability to handle discrete variables, and ability to characterize the entire design space with a minimum of analysis runs. The optimums produced by this method are not true optimums in the sense of numerical minimums. They are, rather, optimums from among the design selected parametric levels. Accurate numerical optimums can be produced through repeated refinement of the grid, but it is more likely that the method would be used early in the design of a vehicle where trends and interactions are the most desired outputs.

All of the design methods discussed in this paper depend heavily on the ability to perform multidisciplinary analysis. That is, given the input variables such as wing sweep, nose radius, etc., determine system outputs such as landing speed, gross weight, and payload capability. This multidisciplinary design process requires that several traditional engineering disciplines work closely to exchange data in an iterative process. This capability exists in such tools as EASIE and AVID. The four design methods discussed here expand on this multidisciplinary analysis capability by providing a structured way to feedback changes in the input variables to improve the design and answer "what-if" questions. The designer as well as the design stand to benefit from the application of these new methods.

Acknowledgements

This material is based upon work supported under a National Science Foundation Graduate Fellowship and by NASA Grant Number NAGW-1331 to the Mars Mission Research Center at North Carolina State University.

References

1. Rowell, L., J. L. Swing, and K. Jones. "Software Tools for the Integration and Execution of Multidisciplinary Analysis Programs." AIAA Paper 88-4448, September 1988.

2. Wilhite, A. "Foundation Techniques for the Development of a Computer-Aided Engineering System for Aerospace Vehicles." PhD dissertation, N.C. State University, 1985.

3. Sobieski-Sobieszczanski, J., and J. Tulinius. "MDO can Help Resolve the Designer's Dilemma." Aerospace America, September 1991.

4. Fox, R. L. Optimization Methods for Engineering Design. Reading, MA : Addison-Wesley Publishing Company, 1971.

5. Morris, S. J., and I. Kroo. "Aircraft Design Optimization with Dynamic Performance Constraints." Journal of Aircraft, December 1990.

6. Kroo, I., J. Gallman, and S. Smith. "Aerodynamic and Structural Studies of Joined-Wing Aircraft." Journal of Aircraft, January 1991.

7. Rogers, J. L. "DeMaid - A Design Manager's Aid for Intelligent Decomposition, User's Guide." NASA TM-101575, March 1989.

8. Barthelemy, J-F. "Engineering Applications of Heuristic Multilevel Optimization Methods." NASA CP - 3031, Recent Advances in Multidisciplinary Analysis and Optimization Conference, part 3, pg. 1029, September 1988.

9. Padula, S., et al. “Demonstration of Decomposition and Optimization in the Design of Experimental Space Systems.” NASA CP - 3031, Recent Advances in Multidisciplinary Analysis and Optimization Conference, part 1, pg. 297, September 1988.

10. Rogers, J. L. “Knowledge-Based Tool for Decomposing Complex Design Problems.” Journal of Computing in Civil Engineering, October 1990.

11. Sobieski-Sobieszczanski, J. "Sensitivity of Complex, Internally Coupled Systems." AIAA Journal, January 1990.

12. Malone, B. and W. H. Mason. "Multidisciplinary Optimization in Aircraft Design Using Analytic Technology Models." AIAA Paper 91-3187, September 1991.

13. Sobieski-Sobieszczanski, J. "Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design : Recent Advances and Results." Journal of Aircraft, December 1990.

14. Barnum, J., et al. “Advanced Transport Design using Multidisciplinary Design Optimization.” AIAA Paper 91-3082, September 1991.

15. McElroy, John. “Dr. Taguchi - Japan’s Secret Weapon.” Automotive Industries, August 1984, pg. 18.

16. Phadke, Madhav. Quality Engineering Using Robust Design. NJ : Prentice-Hall, 1989.

17. Roy, Ranjit. A Primer on the Taguchi Method. NY : Van Nostrand Reinhold, 1990.

18. Hunter, J. S. “Statistical Design Applied to Product Design,” Journal of Quality Technology, October, 1985.

19. McElroy, John. “Experimental Design Hits Detroit.” Automotive Industries, February 1985, pg. 48.

20. Pilon, G. “Product Design Optimization Using Taguchi Methods with Finite Element Analysis.” 1989 ASME Design Technical Conferences - 8th Biennial Conference on Failure Prevention and Reliability, Montreal, Canada, September 1989, pg. 145.

21. Byrne, D. and S. Taguchi. “The Taguchi Approach to Parameter Design.” 40th Annual Quality Congress Transactions, The Taguchi Approach to Parameter Design, 1987.

22. Stanley, D., R. Unal., and R. Joyner “Application of Taguchi Methods to Dual Mixture Ratio Propulsion System Optimization for SSTO Vehicles.” AIAA Paper 92-0213, January 1992.