WL-TR-93-3004

PRELIMINARY STRUCTURAL DESIGN - DEFINING THE DESIGN SPACE

Stephen M. Batill
Richard A. Swift

Hessert Center for Aerospace Research
Department of Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

February 1993

Final Report for Period : September 1989 to December 1992

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-7562

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1993	Final Report: Sep 1989 - Dec 1992

4. TITLE AND SUBTITLE
Preliminary Structural Design - Defining the Design Space

5. FUNDING NUMBERS
CF33615-89-C-3208
PR: 2401
TA: 03
WU: 97
PE 62201F
PE 61101F

6. AUTHOR(S)
Stephen M. Batill
Richard A. Swift

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Hessert Center for Aerospace Research
Department of Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, IN 46556

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Flight Dynamics Directorate
Wright Laboratory
Air Force MaterielCommand
Wright Patterson AFB OH 45433-7562

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-93-3004

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A series of studies were performed to evaluate the application of finite element-based analysis and optimization methods to the preliminary design of lightweight, aerospace structures. Either the internal structural arrangement or the structural materials were used as the design variables in each concept study. The concepts considered ranged from truss structures such as a helicopter tail-boom or complex space truss to a series of semi-monocoque lifting surfaces. Both static and dynamic constraints were used in the design of these concepts.

Multidisciplinary optimization and analysis procedures were used to develop detailed information from finite element models for each structural concept. From this detailed design data neural network representations of the design space were developed for the particular problem. The neural networks were then coupled with either simulated annealing or numerical search optimization techniques to identify improved design concepts.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Structural design, Structural optimization, Neural networks, Simulated annealing, Multidisciplinary optimization, Finite element modeling	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

PREFACE

The work described in this report was conducted during the period from July 1989 to December 1992 under contract F33615-89-C-3208. The research program was entitled "Application of Finite Element Analysis Methods to Flight Vehicle Preliminary Structural Design," and was conducted by the Department of Aerospace and Mechanical Engineering at the University of Notre Dame, Notre Dame, Indiana for the Flight Dynamics Directorate of the Wright Laboratories. The Principal Investigator for the contract was Dr. Stephen M. Batill. The Project Engineer for the contract was Mr. J. David Oetting whose support and comments have been greatly appreciated. The authors also wish to acknowledge the contributions Capt. Robert Canfield and Dr. V.B. Venkayya for their assistance in the installation and operation of the ASTROS system.

CONTENTS

FIGURES

TABLES

SYMBOLS AND ABBREVIATIONS

A	Simulated annealing control parameter variable (Eq. 5.5)
A^*	Rod cross sectional area for redesigned element
$A_{minimum}$	Minimum cross sectional area for rod elements
A_{new}	Rod cross sectional area for redesigned element
B	Simulated annealing control parameter variable (Eq. 5.5)
c, C	Simulated annealing control parameter
C	Local wing chord
CPU	Central processing unit
CTL	Active constraint identifier (Eq. 3.6)
CTLMIN	Violated constraint identifier (Eq. 3.6)
$\mathbf{D}$	Vector of design variables
d	Displacement
E_i, E_j	Objective functions in simulated annealing algorithm
E	Modulus of elasticity, isotropic material or neural network training tolerance
E_1, E_2	Moduli of elasticity, orthotropic material
f(i)	Combinatorial optimization objective function
F_r	Axial force in rod elements
FEM	Finite Element Method
FSD	Fully Stressed Design
g	Constraint function
g_{max}	Maximum value of a constraint function
g_r	Constraint function for fully stressed rod design
g_q	Constraint function
G_{12}	Shear modulus, orthotropic material
i	Index parameter
$\mathbf{i}$, $\mathbf{j}$	Design vectors in simulated annealing algorithm
ICW	Intermediate Complexity Wing
IOP	Input -Output Pairs
k_B	Boltzmann constant
M	Performance index or *Metropolis* criteria parameter
M_{opt}	Optimum value of the performance index
MDO	Multidisciplinary Design Optimization
N	Number of design variables
n	Index parameter
NN	Neural Network
P_c	Probability of acceptance of a new candidate design, simulated annealing
q	Constraint index
Q	Number of constraints
RMS	Root Mean Square
SA	Simulated Annealing
T	Simulated annealing "heat bath" temperature
$t_{Minimum}$	Minimum thickness for planar finite elements, fully stressed design
t_{New}	Redesigned thickness for planar finite elements, fully stressed design
$t_{original}$	Original thickness for planar finite elements, fully stressed design
t/c	Thickness to chord ratio
V_i	ASTROS design variable
w_{ij}	Neuron weights

W, W_{NN}, W_{FSD}	Structural weights
x, X	Nodal coordinate
X_i	Inverse ASTROS design variable
X_{ij}	Neuron input
y, Y	Nodal coordinate
y_j	Sigmoid function "output" (Eq. 4.1)
ΔM	Change in performance index
β	Sigmoid function parameter
δ_{NN}, δ_{FSD}	Linear displacements
Γ	Simulated annealing parameter
θ_j	Neuron bias parameter
ρ	Material weight density
σ_{xx}	Axial stress in rod element or stress component in x direction
σ_{yy}	Stress component in y direction
$\sigma_{(T,C)}$	Tension or compression stress allowable
$\sigma_{(S)}$	Shear stress allowable
τ_{xy}	In-plane shear stress, planar finite elements
ν	Poisson's ratio, isotropic material
ν_{12}	Poisson's ratio, orthotropic material
ω, ω_{iNN}, ω_{iFSD}	Natural frequencies of the ith mode

1. INTRODUCTION AND BACKGROUND

1.1 Preliminary Structural Design

The design of an aerospace vehicle structure is an inherently complex task, with much of the difficulty resulting from the multidisciplinary nature of the problem. Simple analytical or numerical methods and experimentation were used in the past to perform trade studies and provide the basis for design decisions. With the continued development of finite element methods (FEM) and the associated improvements in computing capabilities, designers have gained new and powerful tools to facilitate and accelerate the design process.

The development of the improved structural analysis methods has also stimulated the development of various optimization methods and their application to the structural design problem. These techniques, generically referred to as multidisciplinary design optimization (MDO), now allow for the integration of diverse disciplines, considered simultaneously, into the design process. Typically, these optimization methods, which are based upon a variety of numerical techniques, are integrated with finite element analysis computer software, opening a new dimension of utility for structural design. Unfortunately, these programs are typically limited to the design of a fixed geometry structural model, where the loads, flight conditions, materials, etc., have been previously specified. Ashley[1] states that:

> Optimization has a great potential as a sound way of choosing among alternate concepts, all the way from overall structural configurations to specific materials of construction. Unfortunately, the design of a structure may prove difficult because... the topology and layout of members, boundaries of the volume to be occupied, the choice of material systems, the modes of failure, and the loading conditions likely to be most significant are all matters which are usually addressed as part of the formulation rather than the solution.

The research documented in this report addresses these concerns by exploring a design methodology for effective utilization of MDO design tools at the conceptual and preliminary design stages. These terms are used somewhat interchangeably throughout this report but are intended to imply that phase in the design process where neither the structural configuration nor the specific materials for each of the components have been determined. The methodology involves using the advanced analysis and design capabilities of MDO to provide an analytic definition of the design space for candidate structural concepts. From the design space representation, decisions about structural configuration, material systems, etc., can be made. The ultimate goal of the design process is arriving at a "preliminary" design which satisfies all of the requirements placed upon the design in the most efficient manner. This design would then be subjected to more detailed analysis and component sizing.

Most current analysis software systems have been created in order to quantify the characteristics of a given configuration. In the case of structural analysis, the ability to accurately predict performance is based upon accurate modeling of the structure, its environment and their interaction. In order to perform "analysis" one needs a quantifiable set of parameters which define the system. Design is the process whereby that quantifiable set of parameters is selected and defined. The relationship between analysis and design can be quite complex and it is inherently "iterative."

Most multidisciplinary interactions are not modeled at the early stages of design, where an understanding of the impact of their interactions could lead to substantial improvements in the final designs (or at the very least avoid problems). As an example, consider the wing box of an aircraft. Increasing the wing thickness (t/c) will improve

strength and stiffness characteristics, possibly allowing for the creation of a lighter structure, but this increase in thickness may increase aerodynamic drag. Since both weight and drag relate to range which is often considered as an important design goal, understanding the relative merit of each effect is important. It is important to realize that modification of thickness will also affect a wing's static aeroelastic deformation, fuel carrying capabilities, vibration characteristics, and numerous other characteristics of the aircraft. Determining how wing thickness relates to each of these concerns is a critical issue in design. This simple example implies that not only does the preliminary structural designer require accurate information on a particular design but also needs information related to the importance of certain parameters and their impact on the design. These types of studies, referred to as parametric trade studies, are often conducted in order to provide quantitative insight into the system's "design space." The concepts of the design space and quantifying the design space are central to this research and provided the focus for this project.

It is common when dealing with detailed structural design using finite element methods to deal with highly refined models, with a large number of elements and degrees of freedom. These models are typically able to predict, with good accuracy, the stresses, deformation and dynamic characteristics of the actual structure. If the designer is concerned with a simple stress analysis, results may be easily post-processed into convenient color, shaded contour plots for interpretation. As the analysis complexity increases, involving dynamic analyses, thermal, or aeroelastic phenomena, the amount of data to be interpreted continues to grow. Design, in contrast to analysis, introduces an entirely new dimension to this interpretation difficulty. Specification of appropriate design variables and design constraints must be made. Given a finite element model with fixed internal components, spatial geometry and loading, an MDO procedure will provide a single "point" in the design space. This "design" is often achieved by the selection of the "size" of all or a selected group of the finite elements which were used to idealize the structure. Weight or cost are often considered as the merit functions and the resulting design often referred to as "optimal." Such a point design may provide useful design information, but as mentioned previously this is often not the most useful information for the conceptual or preliminary designer.

Considering that even a relatively simple FEM/MDO analysis and design can produce hundreds of pages of printer output, a means to extract the most significant design information becomes important for efficiency. One possibility for the convenient manipulation, both storage and processing, of design information comes from advances in the area of artificial intelligence. Models based on human brain structure, known as artificial "neural networks" provide an efficient means for storage of quantitative information. These models can be used to interpolate or extrapolate from the known information. They can also be used to provide an effective "mapping" of a structure's design space. It would be advantageous to exploit these neural network characteristics in the preliminary structural design environment.

Once a design space mapping has been obtained, variations in the structural characteristics to perturbations of the design variables can be determined. Values for the design variables may also be determined that will improve the desirable design characteristics. With this perspective the problem at hand can be considered as being composed of a sequence of steps. The first is the determination of quantitative information or knowledge about a specific design, this can be the result of analysis or even experiment. The second is the modeling of the design space, in the current work using the neural network. The third step would be the use of the quantitative model, i.e. the neural network, to provide the information necessary to select the final design concept.

The goal would be to develop a design space representation which provides an effective mapping of the hypersurfaces of the feasible design space (where the feasible design space is an n-dimensional "volume"). The dimension of the hyperspace relates to the number of design variables, and the hypersurfaces represent the boundaries between the

feasible and infeasible design space. The selection of the design variables is an important aspect of this problem. In the case of preliminary structural design, the design variables are not the "element gages" but are instead the basic structural arrangement, the location of the primary load carrying components and the materials to be used in each component.

In the research documented in this report the basic quantitative information about specific designs has been developed using finite element based, MDO procedures, therefore each individual design represents a "best" design for a given structural arrangement and set of materials - for a limited set of "constraints." When this information is used to develop the design space representation, the design space is then based on "optimal" designs obtained from appropriate MDO procedures. Once trained, the neural networks may be incorporated into numerical optimization procedures so that improved designs can be obtained from the design space representation. This approach is particularly useful when the number of design variables is large, and the means to visualize and interpret the design variable interactions are unavailable. As will be shown, this design approach allows for the determination of designs that would otherwise be unobtainable from current design procedures.

It was pointed out earlier that a hyperspace can be imagined for which each point in that space represents a possible design. Also, imagine that measures of performance as well as constraints would constitute hypersurfaces that divide the design space into feasible and infeasible design regions. These concepts can be illustrated by the use of a simple example. In this case the design space is "two-dimensional" so that it can be easily represented in a graphical fashion. A "preliminary" structural design problem representative of the applications to be presented in this report could involve the design of a simple cantilevered five bar truss, shown in Figure 1. Consider the case where it is desired to define the design space for the family of least weight designs as a function of the location of node 2. Understanding how design weight varies as a function of this variable could lead to the determination of a node 2 location that provides the truss configuration that both satisfies all the applicable design constraints and also represents a "least weight" design.

Figure 2 illustrates the variation in truss weight as a function of node 2 location[2]. This plot was generated from graphically contouring the design space defined by 1681 fully-stressed designs. These designs were achieved by incrementing each of the design variables, the node 2 coordinates, through a range of allowable values. This resulted in an even distribution of designs through the two-dimensional design space. Again, the contours shown represented the fully-stressed design weight for the corresponding truss geometry. Similar contours could be generated for other structural characteristics, such as natural frequency, or nodal displacement. The relative simplicity of the structure allowed for the calculation of the least weight for each design in a reasonable amount of time. The development of 1681 individual finite element models and the subsequent "optimization" of each would be prohibitive for a more realistic structural design problem. Throughout this research project, relatively simple truss structures, and their "two-dimensional" design spaces have been used to help provide visual insight into the problems being considered.

From the contour plot, one can see that the lowest weight design can be achieved by situating the node 2 location in the lower left corner of the design space (specifically, at X_2=7.0, Y_2=0.0). This determination can be made from a simple visual inspection (your own highly developed neural network, pattern processing) of the contour plot, but what happens when the dimension of the design space increases beyond our ability to visualize it? In what form can the design information be represented, and once represented, how can determinations about the structural configuration to best meet a set of design objectives be made?

A final complicating issue deals with design variables that are not continuous valued (as they are in the previous example) but, rather, discrete valued. This could be represented by having multiple materials under consideration for each of the five rods. Consider again the five bar truss, this time with two material possibilities for each of the five rods. This

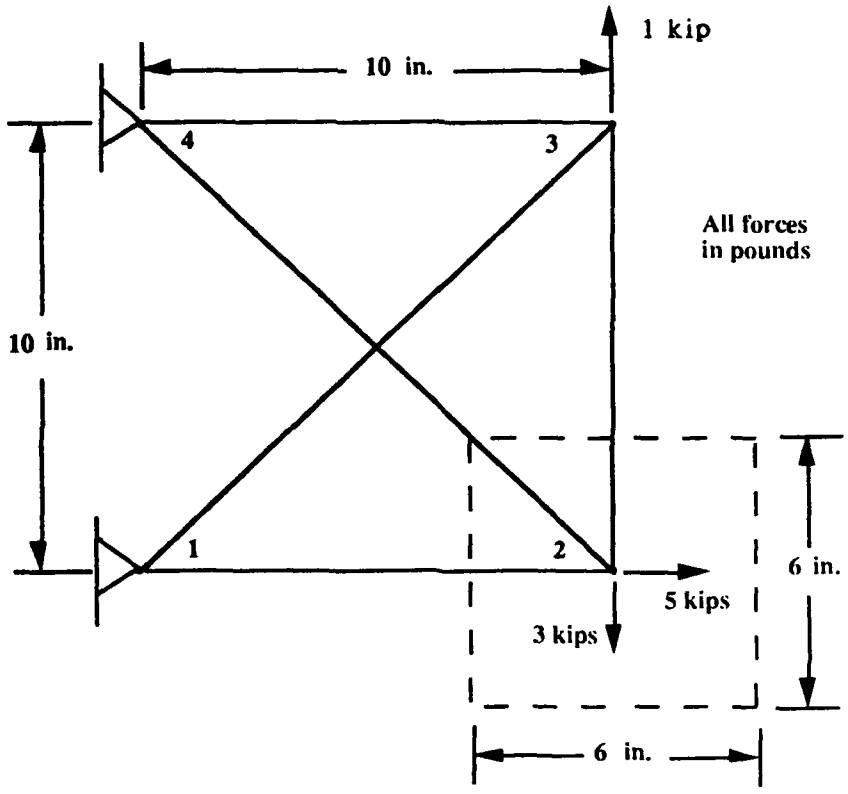


Figure 1. Five Bar Truss, Simple Design Problem

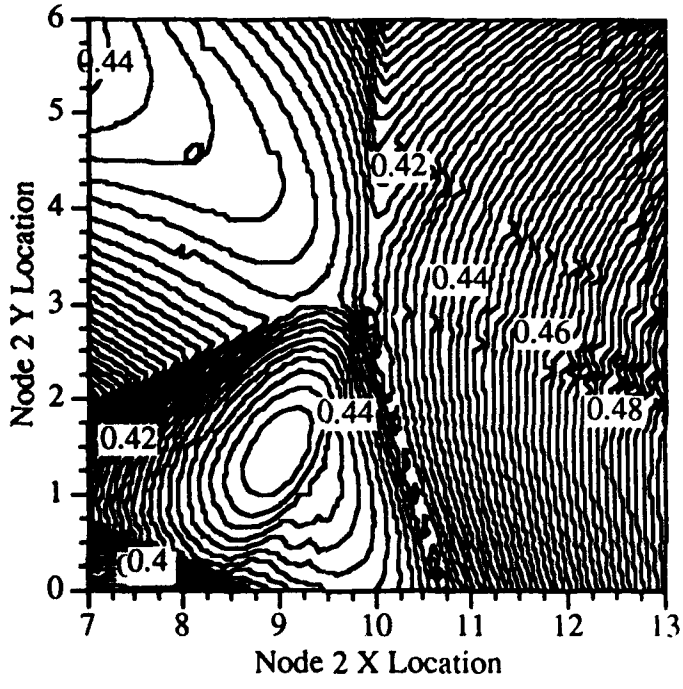


Figure 2. Weight Design Space for the Five Bar Truss - Configurational Design

represents 32 possible material combinations. The materials under consideration are shown in Table 1. Figure 3 shows the results for the discrete design space representation. The 32 designs were obtained from using the baseline geometry of the five bar truss and the loading shown in Figure 1. These designs were determined using the Automated Structural Optimization System (ASTROS)[3] which employs math programming methods to develop the least weight structures subject to various constraint conditions. ASTROS was used as the source of design information for many of the design studies conducted during this research and is discussed in some detail later in the report.

The horizontal plane of the Figure 3 represents the material composition of each of the rods. One of the horizontal axes represent the eight unique material combinations for rods 1, 2, and 3, respectively. The other horizontal axis indicates the composition of rods 4 and 5. One can see that the trend is towards a reduction in weight as the number of titanium rods is increased. The least weight design is not solely composed of Ti, however. A material assignment vector {2,2,1,2,1} for the five rods yielded the least weight design. The final design areas for the five rods was {A} ={ 0.04, 0.013, 0.01, 0.037, , 0.01} in^2. Two of the elements were at a minimum gage constraint of 0.01 in^2 and this is the reason why an all Ti design is not the least weight design. This particular example helps illustrate the complexity associated with a "combinatorial" optimization problem which is not characteristic of the continuous design space represented in the first example presented above.

The two previous examples have presented representations of the design space for two specific design problems. The tabular data which was used to represent the design space in a graphical fashion could also be used to provide the information necessary to "train" a neural network. The neural network can then form an approximate representation to the n-dimensional design space (discrete or continuous) under consideration. The representation is approximate because it is based on a set of "point designs," from which the global representation is obtained. The neural network allows for an interpolation or extrapolation from the existing design information to obtain this complete n-dimensional design space representation, much as the contour plotting procedure does a fit to the points of data from which it is composed. The neural network representation then provides an extremely efficient, quantitative means to investigate the design space characteristics of the structural design problem under consideration.

1.2 Background : Multidisciplinary Analysis and Artificial Neural Networks

The following is intended to provide a basis of understanding for some of the major concepts and techniques described in more detail later in the report. A brief description of multidisciplinary design optimization (MDO), fully-stressed design and neural networks for use in design space synthesis is provided. Discussion regarding previous research in the area of neural network applications to structural design is also outlined.

In determining a structural design, a single configuration (materials, structural geometry, component sizing) can be selected, based on some established criteria, from those candidate configurations that are acceptable under the appropriate constraints. To obtain this structural configuration, the designer uses his ability to predict the structural behavior of the model under a given load or loads and his previous experience with similar design problems. At best, this approach can only provide clues to the best solution, not its identity.

Given a complex structure which can define an equally complex design space, what approaches can be taken to extract the maximum amount of design information? Also, since the conceptual or preliminary design stages are inherently where major inter-disciplinary trade-offs are made, how can the major trends of interest be extracted from the wealth of information obtained from the various designs? Simplified finite element models of

Table 1. Material Properties for the Five Bar Truss

Material	Modulus (psi)	Density (lb/in^3)	Tensile Allowable (psi)	Compressive Allowable (psi)
1) 7075-T6	$10.5(10^6)$	.101	$36.(10^3)$	$37.(10^3)$
2)Titanium	$17.0(10^6)$	.164	$75.(10^3)$	$75.(10^3)$

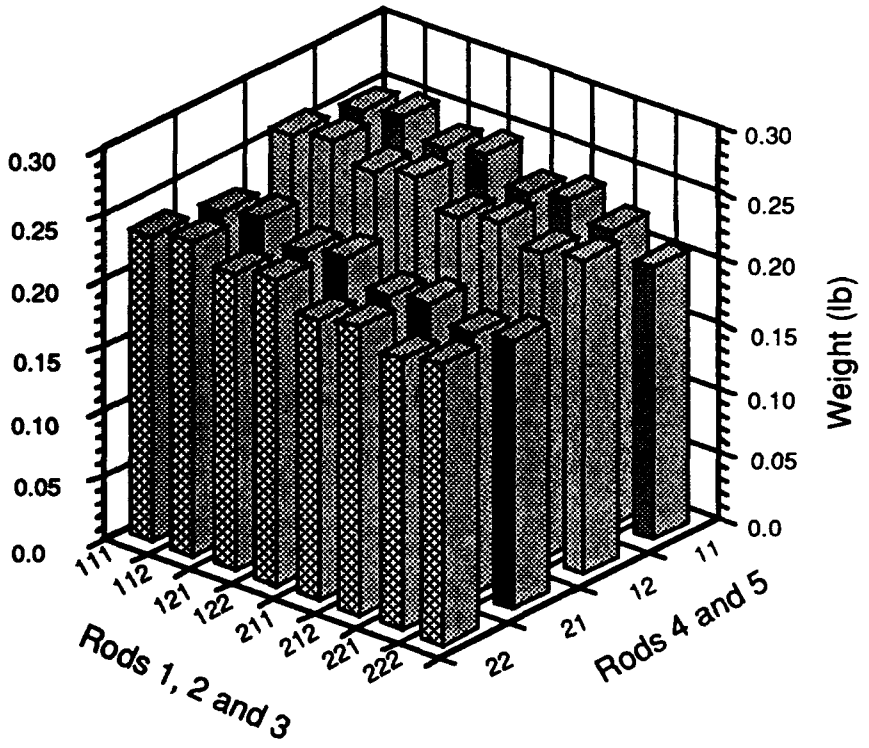


Figure 3. Discrete Design Space for the Five Bar Truss - Material System Design

structures have already been shown to be an effective means for investigating the behavior of complex structures[4]. It is thus desired to formulate simple, yet representative, finite element models of candidate structural concepts and investigate possible structural configurations of interest. A representative model is defined here as one in which the major macroscopic properties of a given structure can be accurately determined. Because these simple finite element models are typically computationally efficient for analysis, numerous models may be analyzed and the resulting design information may be combined to form a "mapping" of the structure's design space. This mapping will contain information such as major characteristics and their variations as a function of the design variables (design sensitivities). One of the major difficulties arises at this point; that is, for even a single, simple finite element model there is a wealth of design information provided by a finite element analysis. Given a set of possibly several hundred finite element analyses, the amount of information may become too vast to be interpreted efficiently.

Work on artificial neural network models began more than 40 years ago, with the development of detailed mathematical models by McCulloch and Pitts[5], Hebb[6], Rosenblatt[7]. Swift[8] discusses in more detail aspects of the recent developments in neural networks and provides additional references. This particular area has received considerable recent interest and the number of papers, journals and books on the subject has increased exponentially. For the research presented in this report, only one type of neural network was employed; a feed-forward neural network that employs back-propagation training. These networks can "learn" from a set of information and provide solutions when "new" information is requested. A more detailed discussion of these networks is presented in Section 4.

Feed-forward networks have a set of inputs that form the "input layer," and a set of outputs which form the "output layer." When neural networks are used for design space mapping, the design problem at hand defines the number of inputs to the neural network as well as the number of outputs. The network inputs correspond to the design variables of interest, while the network outputs represent the objective function(s) or constraint quantities. Hidden layers (intermediate layers to the input and output layers) are required for all but the simplest problems. The hidden layer geometry is a function of the design space complexity to be mapped. Since the neural network representation is not unique, many different hidden layer geometries will provide adequate design space representation. Once a neural network configuration (i.e., number of neurons and connections) is selected, the parameters which define the neuron characteristics must be determined. This is accomplished using the training data or design "knowledge." The training data are sets of information whose functional relationship one wishes to represent with the neural network. This training data is in the form of a set of input/output vectors.

The current research builds upon previous efforts to apply neural networks to structural analysis and design. Rehak, Thewalt, and Doo[9] described the utility of neural networks for computations in structural engineering, in particular dynamic structural system identification problems. The case of a damped two-degree of freedom structural system was considered. This model was subjected to a loading representative of an earthquake response simulation. Training data for the neural network was obtained from a second order ordinary differential equation that defined the behavior of the system. The inputs to the network were acceleration, velocity, displacement, and force at time instant "i." The network output was the expected acceleration at time "i+1." The network was able to perform the dynamic system modeling successfully. The authors also described the influence of the parameters used in training the network on learning effectiveness, and the response to various input vector schemes. Simple feed-forward networks without hidden layers were considered. A number of major conclusions were drawn from this paper, the most significant being the promise that neural networks showed for a wide range of difficult engineering problems.

Hajela and Berke[10,11] demonstrated the utility of neural networks as a substitute for the finite element analysis procedure in an optimization analysis. This work involved training a series of neural networks to represent the force-displacement relationship developed from a static structural analysis. The trained neural network effectively replaced the finite element analysis portion of a math-programming optimization procedure. As an example, they consider a five bar truss with a fixed geometry. Random values for the cross-sectional areas of the five rods were determined, and a static analysis of the truss was performed. These rod areas and the resulting nodal displacements were used as training data for a neural network. Stress values were calculated from these displacements. It was determined that the neural network could effectively replace the finite element analysis portion of an optimization procedure, where the design variables were the rod cross-sectional areas. The effects that neural network geometry had on the force-displacement representation accuracy and on the required network training times were investigated. Their results showed the potential for further applications of neural networks to design.

Swift and Batill[2,8,12] have used neural networks to map out the configurational and material design spaces for a number of structural design problems. The work presented in Reference 2 and 12 involved training a neural network using training data obtained from finite element analysis where nodal locations and material properties were considered as the design variables, respectively, rather than just element gages. The neural networks were able to effectively represent the design information, and optimal configurations were obtained to meet constraints on weight, displacement, and natural frequency for simple trusses and a built-up wing-box model. The neural networks also displayed the ability to isolate regions of the design space where improved designs were residing, resulting in significantly improved final designs. Reference 8 presents a comprehensive development of the background for many of the topics presented in this report.

Swift and Batill have also explored the utility of neural networks for damage tolerant design[13], and have looked at a recursive learning procedure[14]. The recursive learning procedure is an attempt to address the minimization of training data generation for effective design space representation. The generation of training data is typically the most expensive part of the neural network procedure, since finite element methods and math-programming procedures are used as the source of this information. These procedures are computationally intensive, and require significant computer resources and time investment to generate training data sets.

There is a growing interest in the use of neural networks for engineering design applications. Carpenter and Barthelemy[15] have performed a comparison between the ability of neural networks to represent information in contrast to common curve-fitting techniques. It was shown that the neural networks offer a more flexible means for the representation of complex data sets that have variable input/output dimensions. Fu and Hajela[16] have employed Hopfield networks for the minimization of distortion in truss structures. Applications of neural networks to other aerospace problems have become too numerous to list. One notable paper is that of Linse and Stengel[17], who have used neural networks to formulate a "smooth" aerodynamic model for use in adaptive nonlinear control strategies.

A number of conclusions can be drawn from the previous work utilizing neural networks for engineering applications:

1. Neural networks provide a flexible means for performing calculations that are difficult to formulate or are otherwise computationally costly, and

2. With the current shift towards implementing neural networks in hardware (rather than in software) very significant improvements in the computational times required for some types of computations can be expected.

The focus of the research presented in this report attempts to utilize the favorable neural network characteristic--employing neural networks for the difficult task of computing design information (specifically design space representations based on both discrete and continuous design variables). Considering the continued development of

neural network technology, the development of methods and procedures that allow for the application of the neural network modeling of non-parametric design spaces may have important future benefits.

1.3 Overview of the Current Research

The application of MDO procedures and neural networks to preliminary structural design has been applied to two categories of design problems, each relating to a different design variable (and design space) "character." The first is that of designs involving continuous design variables (continuous in that the variables may have any value between an upper and lower bound constraint). The second category considers discrete design variables. These discrete design variables represent such things as material type (assigning specific materials to specific structural components). In both cases the goal was to develop a means whereby the advantages presented by the detailed information provided by the MDO procedures could be effectively integrated into the structural design problem prior to the selection of the final configuration or materials for a structural concept.

For the cases using continuous design variables, two sets of problems are presented. Both sets involve configurational design. That is, determining the structural configuration that provides the most desirable design characteristics. These two cases involve the design of a ten bar truss and a three spar wing box. For the ten bar truss, the spatial location of three nodes were allowed to vary within specified regions. This allowed for a wide range of possible truss geometries. The weight design space for the least weight trusses (based upon a fully stressed design concept) were represented using a neural network for this six-dimensional design space. This example was intended to highlight the major considerations involved in the neural network design space mapping procedure. An optimal truss configuration to minimize weight was determined. The second example in this set was that of a three spar wing box, in which spar locations are considered as the design variables, and the design space defined by weight and natural frequency. These were then used as the merit functions in an optimal design study which then used the neural network in lieu of the more costly MDO procedure. This configurational wing design problem is intended to describe a more complex, and realistic, application of the neural networks for design space representation.

The second set of configurational design problems involved the use of recursive learning, which is a procedure used to minimize the amount of data generated to form the neural network design space representations. Reduction of training data was useful since training data generation involving FEM and MDO procedures was very time consuming. Recursive learning effectively refines only those regions of the design space that show the most promising design characteristics. Two examples are presented in this report. The first is another ten bar truss, this time with only a single node considered for the configurational design. This simple example describes the basic implementation of the recursive procedure, and allowed for a "visualization" of the design space as the recursive training of the neural network proceeded. The second example involves the configurational design of a four spar wing box, where weight, tip deflection, and natural frequency are considered independently as the objective functions for a subsequent configurational design using the neural network model for the design space. Again, this wing box example was meant to represent a more realistic application to configurational design and the recursive learning approach.

The case of discrete design variables was considered next. A series of examples are presented. In order to be consistent, the first example is that of a ten bar truss in which four materials were considered for each of the ten structural elements which made-up the truss. This example is used to show the applicability of the neural network approach to discrete design space representation, as well as the ability of the neural network to extrapolate from a given set of designs to determine improved designs. For the design problems involving

discrete design variables, the neural networks were then used in conjunction with a simulated annealing procedure to determine least weight material configurations.

The second discrete design variable study involved the material design of an Advanced Control of Space Structures II[18] (ACOSS II) space truss, in which four materials were considered for each of the 113 structural elements (a difficult combinatorial optimization problem with 4^{113} possible material systems). An "estimate" of the time required to perform an exhaustive search for this combinatorial design problems was 10^{52} years using conventional finite element analysis and optimization. This example showed the utility of the neural network approach to the design space representation of extremely large discrete design spaces based on very small (small in relation to the combinatorial size of the problem) training data sets. Material systems that minimize the weight of the truss were determined by using the trained neural network as a design space approximation for the simulated annealing procedure. The simulated annealing procedure provided the means to traverse through the discrete design space in an efficient manner to determine improved designs.

The final discrete design variable case involves a brief study of the use of this type of design variable modeling to the skin design of a composite wing. An alernative method for the designation of the material design variables was considered in this case.

The last set of design studies involved the specific design application of MDO and neural networks to the problem of damage tolerant structures. For this application case two structural concepts were considered. The first was the configurational design of a helicopter tail boom for which six possible damage conditions were considered. A fully-stressed design procedure was used to achieve minimum weight designs while satisfying all applicable stress constraints. Weight and natural frequency were considered as the objective function/constraints for the neural network analysis. The neural network was able to accurately represent trends and identify optimal configurations. In the second example, an undamaged baseline five spar wing box was designed for flutter and stress constraints. One hundred and fifty possible damage states were analyzed for flutter and natural frequency characteristics. These results were used to train a neural network. The neural network was then used to predict flutter occurrence and the natural frequencies for all possible damage conditions.

For all the examples, both continuous and discrete, various network configurations and levels of training data were employed. This helped provide some insight into both the character of the structural design spaces as well as to evaluate the use of neural network modeling of the design space. The following sections of this report are devoted to a discussion of the MDO procedures used during this study, a brief discussion of neural networks including training procedures. The remainder of the report is then devoted to a presentation of each of the design studies mentioned above and presents information on the models, analysis methods and results.

2. STRUCTURAL MODELING FOR PRELIMINARY DESIGN

This section provides a brief overview of a number of aspects of the preliminary structural design problem based upon finite element modeling and analysis. The section focuses on the those issues related to the preliminary structural design of lightweight structures with a particular emphasis on lifting surfaces.

2.1 Definition of Design Variables

The process of structural design is complex and not well suited to being portrayed as a specific process. Usually the design of an airframe or of a particular component is an evolutionary rather than a "revolutionary" process. One attempts to take advantage of successful aspects of similar earlier designs and to avoid the problems encountered in previous "failures." The eventual purpose is to develop the guidelines (i.e. drawings, CAM instructions, etc.) which will allow for the fabrication of the structure.

The structural design usually takes place at a point in the flight vehicle design process where the aircraft performance requirements and external aircraft geometry have been defined. These conditions are used to provide the "mold" lines for the structure and the loads, temperature, etc. to be used in the selection of the materials and their distribution within the "structure." Given these requirements the designer then must select the material, size and relative orientation of each of the structural components. Those specific parameters over which the designer has "control" are referred to as design variables. The thickness at a particular point on a spar, the number and distribution of rivets, and the skin material are design variables. Weight, natural frequency, deflection at a point or flutter speed are performance indicators or measures of merit but they can only be indirectly controlled through the selection of the design variables.

In the past structural design has played a somewhat limited role in the basic aircraft conceptual design phase. The basic aircraft geometry was defined based upon "aerodynamic" considerations and thus became a constraint for the structural designer. Often target weights were selected based upon pervious designs and these too served as design goals subject to the wide range of performance constraints.

As the structural design proceeded, loads were defined and the basic structural concept selected. An iterative process was initiated in which static loads analysis and design typically preceded other considerations such as fatigue, aeroelasticity(static and dynamic), survivability, reliability, manufacturing, etc. The structural designer worked to achieve acceptable margins of safety on all components for a wide variety of flight conditions. Satisfying these often conflicting requirements has always been a challenge to the structural designer, particularly considering the extreme emphasis placed upon weight for almost all flight vehicle applications. As this iterative process developed, those parameters referred to as design variables also change. Since the mold lines may be fixed, they are not acceptable design variables and are used in the formulation of the structural design problem, not its solution. When the basic structural layout is selected, such as the number and placement of primary structural components (i.e. spars and ribs), these parameters are no longer variables. New variables, such as the size of these components, are added. As new details are included, new design variables must be determined along with the sensitivity of the design to these parameters.

Recently more emphasis has been placed upon the integration of the various technologies earlier in the vehicle design process with the expectation that this can allow for improvements in both the design process and the product. This has allowed the designer to work with a much larger variety of design variables. In most cases as more variables are introduced into the problem, the more complex and time consuming it can become. The proper selection of design variables is an important consideration at each phase in the process. It appears as if the longer a parameter can be "active" in the design

process, the greater the benefit that can be achieved by the proper selection of that parameter. As one might expect, the ideal situation would be to let every variable participate in all levels of the design, but the obvious practical considerations make this impossible.

The goal of improving the preliminary structural design process has been the motivation for the research presented in this report. As mentioned in the previous section, the ability to use the best available analysis methods as early as possible in the design process when the smallest number of design variables have been fixed may allow for the structural designer to take a more proactive instead of reactive role in the system design process.

In this report a number of different types of design variables are used as part of the analysis and design studies. They are briefly discussed below with emphasis placed upon the challenges they bring to the preliminary design process. The term "design variable" is used to describe those parameters which define the characteristics of a "model" of a structural concept. Given a complete set of these parameters, appropriate analysis procedures can be used to determine the characteristics of the model, from which one can infer the performance of the structure.

2.1.1 Continuous Design Variables

These may represent the most common of the design variable types. The design variable can be used to define a parameter which can vary continuously, consistent with the appropriate level of numerical discretization, over some range of values. Examples of this type of variable might be the thickness of a metal wing skin at a given point, the chordwise location of a spar or the diameter of a hole. Though cost may be a factor, there are no practical or physical constraints that would not allow this value to assume a given value, plus or minus a prescribed tolerance.

This type of parameter is often used in analytic optimization procedures. If various measures of merit can be expressed either analytically or numerically as functions of this variable, numerous methods exist which can be used to identify local or global extreme values of these measures of merit. These extremum can be used to provide information necessary for design decisions. These parameters are often used in "parametric" trade studies to help describe the behavior of the system for a range of the parameter. This data is presented on a "carpet plot" in order to provide a visual description of the influence of design variable on the measure of merit.

2.1.2 Discrete Design Variables

The second type of design variable is one which is only allowed to assume a finite number of discrete values over some prescribed range. Examples of this type of variable would be the number of composite plys in a wing skin, the number of spars, or the number of rivets in a given rivet pattern. These variables can be usually designated as a series of integer variables, though the sequence of the integers may have little physical significance.

This type of design variable definition leads to a combinatorial design or optimization problem. Since there are a finite number of design variables which can assume only a finite number of values, it is conceivable that an exhaustive search of the design space would be possible. This is true only for problems of very limited size and complexity; however the cost of analyzing all of the potential designs becomes prohibitive very quickly .

Another aspect of this type of problem is the discontinuous behavior of the merit function. This presents problems for conventional numerical optimization methods as well as introducing difficulties, even with visual presentation of this type of information.

2.1.3 Configurational Design Variables

In this report a number of configurational design variables have been used. These design variable have been used to represent parameters which provide the relative spatial arrangements of components within a structure. As will be discussed below, these types of variables have a large influence on the numerical model used to represent the structure.

Configurational design variables could be continuous, such as the chordwise position of a spar, or discrete, such as the number of spars. In either case, the parameter is related to the geometry of the model used to represent the structure. Since significant effort is often required to generate the models used for structural analysis, changes in those variables which influence the basic model can be rather "costly." Thought these types of design variables are not often used in conjunction with finite element analysis, there has been more emphasis placed on configurational design in recent years.

2.1.4 Material System Design Variables

In addition to the relative orientation and size of a particular structural component, the material used to fabricate the component is an important design issue. Material parameters such as stiffness, density, allowable stress and strain levels are just a few of the material characteristics used to select an appropriate material. In the design problems considered in this report, material selection has been developed as a discrete design variable problem, where the designer is allowed to select from a limited list of potential materials with prescribed properties. This should be differentiated from issues related to the design of a material for specified performance. The design of a material with specified performance was not considered in this study.

2.2 Finite Element Based Structural Analysis

The basic analysis procedures used in this study to determine the characteristics of a structure were based upon the finite element idealization and associated displacement method of structural analysis. There are numerous analysis methods available but the two outlined in the following sections were selected for this study. The first software system, SWIFTOS[19], was selected since it was developed in-house and therefore it was possible to modify it for the series of special problems considered in this study. SWIFTOS is a rather small and efficient code for relatively simple analysis and design problems. The second software system, ASTROS[3], was selected since it represented the state-of-the-art in finite element based multi-disciplinary analysis and design. Each of the methods provided the design space information necessary to perform the preliminary structural design studies in this report.

2.2.1 SWIFTOS

SWIFTOS[19] (an acronym for the Structural WIng Optimization System) was one of the two finite element analysis and design packages used for the generation of design information in this study. SWIFTOS was created for the preliminary static stress analysis and fully-stressed design (FSD) of either space trusses or planar lifting surface structures.

The code is based upon the displacement method and has a very limited library of element types. Pinned truss structures were modeled by rod (axial force) elements and only fixed or free boundary conditions were available at each node. The lifting surface models were developed using subparametric quadrilateral membrane elements (plane stress) and rod elements. Only simple distributed pressure loads or point loads were used in these idealizations and a dynamics analysis capability based upon the consistent mass matrix representation was included. SWIFTOS provided information on nodal

displacements, element stresses, natural frequencies and finite element model weight in a small and efficient analysis package. Additional details on the element properties and the methods of analysis used in SWIFTOS can be found in Reference 19.

2.2.2 ASTROS

ASTROS is a more general design package, developed by Northrop under contract to the Air Force, which incorporates many of the finite element types of NASTRAN[20]. Reference 3 is a series of documents which provide the theoretical background, user instructions and sample case studies using ASTROS. The software contains a rather extensive element library although only a limited number of element types were used in the studies documented herein. ASTROS was used for both its analysis and optimization capabilities.

ASTROS also has aerodynamic paneling techniques for load generation and a large range of applicable design constraints. It has a large range of analysis capabilities along with an associated larger "computational overhead." In ASTROS, the user can specify for output a wealth of intermediate as well as final calculation data. Though a brief discussion of the use of ASTROS as a source of design space information is provided in a later section, the reader is encouraged to examine Reference 3 for more details on the capabilities of ASTROS.

2.3 Automated Finite Element Model Generation

The ability to rapidly and efficiently develop the finite element representation of a candidate structure was an important consideration for this research. If this process was to be conducted by hand, even a relatively crude finite element model for a lifting surface could require "man-days" to formulate. Because large numbers of finite element models were to be created for this research, a data preprocessor, called XPUT, was developed. This FORTRAN software used a non-graphical, text interaction with the user to develop both the finite element model data as well as the other information necessary to perform structural analysis or design of a specific concept. The information provided was similar in form to the data available to the structural designer such as airfoil section geometry, numbers of ribs, spars etc.. This was then converted into the necessary information and formatted for the finite element analysis. Additional details on the basic model descriptions are provided below.

This preprocessor had the capabilities to be operated in an interactive mode with an informed user to define the finite element model for a specific design concept. Additionally, the ability to operate in an "automated" fashion was included. That is, those model characteristics or design variables, that were to be modified, along with the modification scheme, were input to the preprocessor and the finite element models were generated automatically. This became especially important in the recursive learning cases, where the neural network provided "feedback" to the preprocessor -- in effect specifying the geometry or material types for the next finite element model.

The two main families of models considered for this research were simple pinned trusses, and built-up semi-monocoque wing box models. The development of these model types will be discussed in the following sections.

2.3.1 Finite Element Model Types

The discussion of model types will center on simple planar truss models used for the demonstration design studies and the semi-monocoque finite element wing models used for the more involved design studies. These models were created using XPUT or a modification of XPUT. The ACOSS II space truss used for material design was obtained

from Reference 3. This space truss model's basic configuration or element arrangement did not vary through the study so it did not involve the use of XPUT.

The truss models were created, analyzed, and designed using a modification of the basic SWIFTOS source code. This modification involved the utilization of only the specific SWIFTOS subroutines needed to model the truss under consideration. As an example, for the five bar truss example shown in the Section 1, only the rod element stiffness subroutine, the global stiffness assembly subroutine, and the simultaneous equation solver were used for the analysis portion. This group of subroutines was in turn used by the fully-stressed design procedure to obtain the least weight designs. The XPUT preprocessor was removed for the truss problems, and in its place was substituted a simple model geometry sequence -- a sequence that was unique for each truss problem considered. This model geometry sequence was nothing more than a description of the nodal coordinates, and how these coordinates were allowed to vary when being considered as the configurational design variables. After each new truss was created by the geometry sequence, it was designed using the FSD procedure. The finite element model and design information were then stored for later use. This truss design procedure was very efficient, with the ability to literally design hundreds of trusses in a matter of minutes.

This illustrates the utility of a procedure like SWIFTOS, though similar studies could have been conducted with ASTROS through the use of a special executive sequence, since the purpose of each of the two software packages was to provide preliminary design information, the most efficient package was used for each design study considered.

The planar lifting surfaces under consideration were of a conventional semi-monocoque design, and had as their main structural components spars, ribs, rib and spar caps, and upper and lower skins. The configurations for the models generated by XPUT were the same regardless of whether the final output was in SWIFTOS or ASTROS format. A simple but representative finite model used to analyze this type of lifting surface would resemble the model shown in Figure 4.

The finite element model consisted of a set of quadrilateral membranes representing the skins, spar webs, and rib webs. Rod elements could be used to represent the spar caps, rib caps, and connecting posts (which connected the upper and lower surface nodes). In SWIFTOS, subparametric quadrilateral membranes were used to model the membranes while in the ASTROS implementation of XPUT, spar webs, rib webs, and skin elements could be represented by either isoparametric quadrilateral membrane elements or shear panel elements.

To define the finite element model, the following information was provided to the preprocessor:

1. Root and tip chord for the planform
2. Lifting surface semi-span
3. Sweep and dihedral angles
4. Airfoil cross-section
5. Lift distribution information - (Magnitude and distribution for the SWIFTOS version or dynamic pressure, air density, Mach number and load factor for the ASTROS version)
6. Material properties
7. Specification of element groups (as well as element types in ASTROS version)
8. Specification of element materials
9. Specification of element thicknesses and cross-sectional areas
10. Specification of design variables

As one can see this is not the typical information provided to a finite element analysis program. The goal was to develop the capability to obtain the required finite element data from information more typical of the structural designer's vocabulary. This was then translated into the necessary format for either SWIFTOS or ASTROS. This list

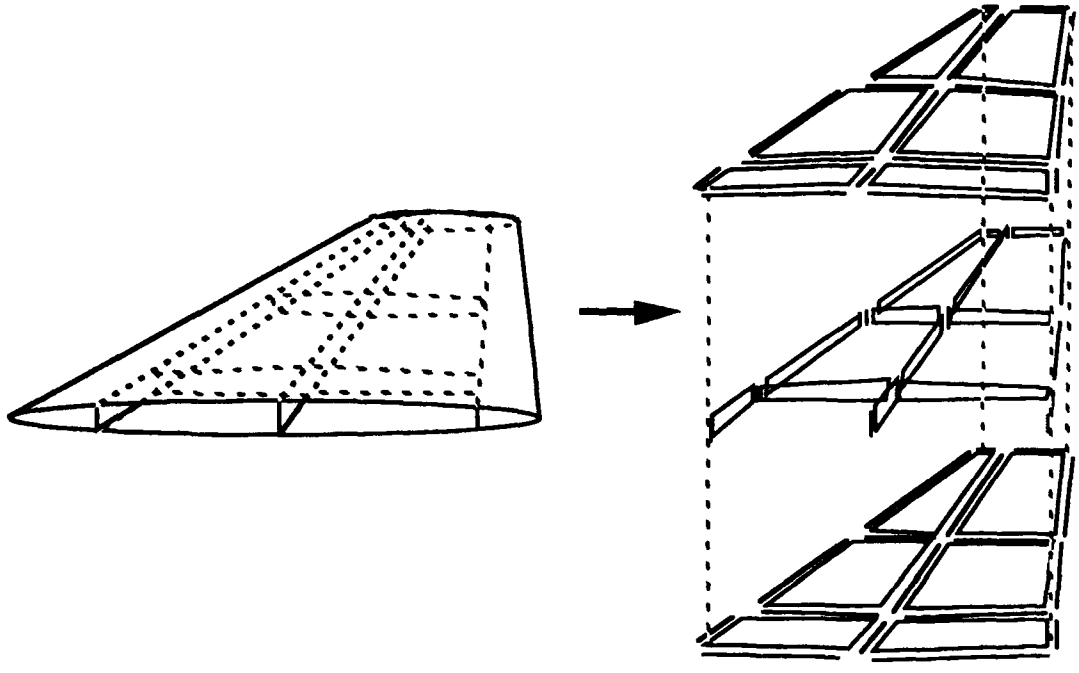


Figure 4. Typical Finite Element Representation for a Lifting Surface

provides the basic physical parameters necessary to define a lifting surface structural concept.

This information provides all the necessary data for the formulation of the finite element model. A cantilevered boundary condition was applied in SWIFTOS (cantilevered at the root), while in ASTROS an "inertially supported"[3] boundary condition is applied. An experienced XPUT user could create a complete finite element model in several minutes, a model that was ready to be analyzed or optimized. While the resulting models were intended to have a high degree of flexibility in their configurations, the formulation was limited by a number of constraints. These constraints defined the range of possible model configurations, element types, and loading conditions. When operated in an automated fashion, the generation of each finite element model took a matter of seconds. Figure 5 illustrates the level of detail for a typical lifting surface finite element model created by the pre-processor and used in this study. This model had 3 spars at 10, 35, and 85 percent chord locations. There were 13 ribs evenly spaced in the spanwise direction. The full aerodynamic planform is shown by the dotted lines in the Figure 5. The model was composed of 64 rods and 117 membranes with 234 degrees of freedom (three degrees of freedom at each node). The wing was cantilevered at the root.

For the SWIFTOS designs the subparametric elements were used for the skin and spar and rib web elements. These subparametric elements based upon the development in Reference 21 provided reasonable in-plane bending stiffness. The subparametric elements showed good correlation with similar models developed using ASTROS in which the isoparametric elements were used for the skin and shear panels were used for the rib and spar webs. For each of the design studies documented later in the report, the same model type and analysis procedure was used so that all comparisons were made for a similar idealization.

2.3.2 *Applied Loads*

A number of aerodynamic loading schemes have been used in finite element analysis packages, many of them quite accurate in their modeling of the aerodynamic forces (and just as complex). ASTROS employs USSAERO[22] for subsonic and supersonic steady aerodynamics, Doublet Lattice Method[23] and the Constant Pressure Method[24] for unsteady aerodynamics. Simplified but realistic distributions were considered acceptable for SWIFTOS, and it has been shown that a distributed pressure loading may be adequate at the preliminary design stage[25].

The preprocessor, XPUT, was used to create a simplified pressure load for use with both SWIFTOS and ASTROS. The pressure load applied over the lifting surface is termed the "loading plane." The magnitude of the pressure load could be varied in both the chordwise and spanwise directions, as specified by the user; this distribution could be adjusted to give the desired total load and center of pressure for the corresponding flight condition. After the loading distribution was determined, the planform was divided into regions, termed "aero-panels." The pressure over these panels was then integrated to determine a set of discrete point loads, which were then transferred directly to the model nodes. Figure 6 shows a typical loading plane used for a SWIFTOS load distribution. For a detailed description of the SWIFTOS loading development refer to Reference 19.

As mentioned previously, the generation of the aerodynamic loads in ASTROS followed a discrete paneling scheme where information on airfoil cross-section, wing planform, and flight conditions was used to determine the pressure distribution over the lifting surface. This pressure distribution was integrated and splined[26] to the structural nodes of the model. The calculation of the aerodynamic loading was typically a computationally expensive portion of the ASTROS design procedure, and use of the simplified pressure distribution appeared to be an acceptable alternative based upon computational efficiency at the conceptual design stage.

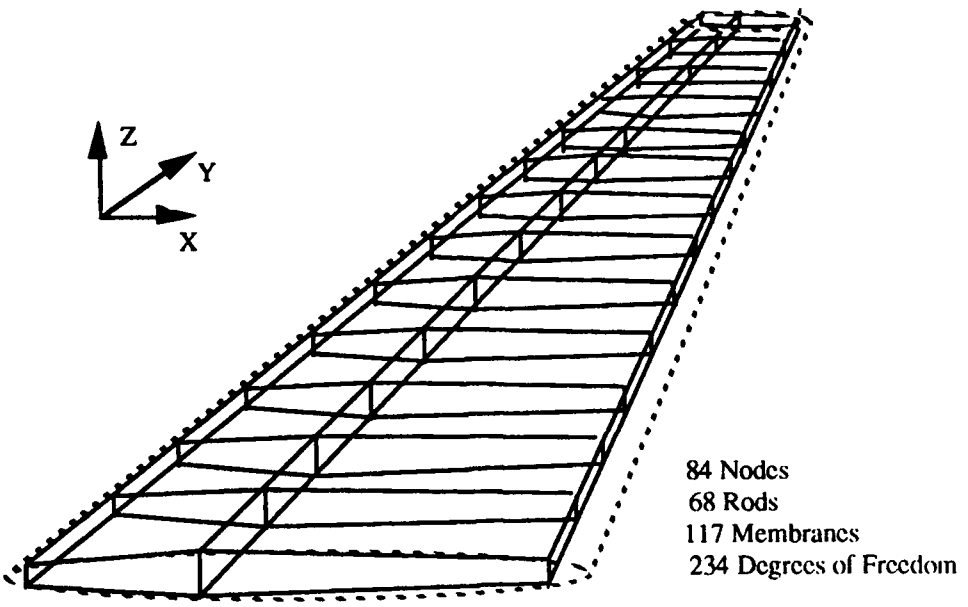


Figure 5. Sample Lifting Surface Model

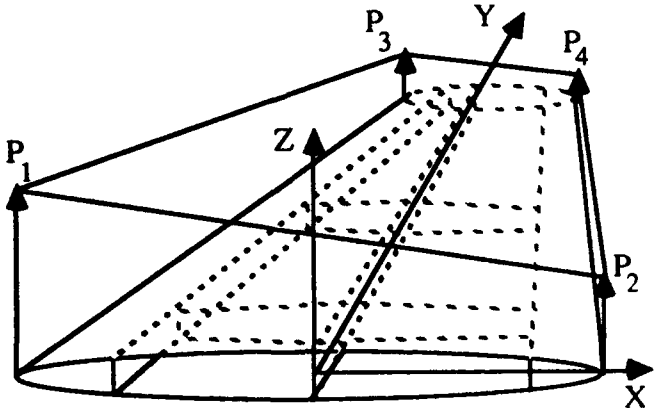


Figure 6. SWIFTOS Aerodynamic Load Model

2.3.3. Material Properties Specification

The SWIFTOS finite elements used isotropic materials, requiring that the modulus of elasticity (E), Poisson's ratio (ν), weight (or mass) density, as well as the maximum allowable tensile, compressive, and shear stress values be input. ASTROS can use isotropic, anisotropic, and orthotropic materials, but XPUT only considered isotropic materials in the automated preprocessing. For the applications considered here, up to five materials were used in a finite element model. This was a limitation of the XPUT procedure and not a limitation of either SWIFTOS or ASTROS. E and ν were used in the stiffness generation procedures, while the allowable stress values were used in the design sequence. The material density was used to calculate the weight (or mass; this was simply a density times volume) of the model.

Individual element groups could be composed of the same material, or of different materials. The material element groups were defined as follows:

1. Individual spars
2. Individual ribs
3. Skin bays (defined by those skin elements that connect two adjacent spars)
4. Connecting posts (all)
5. Spar caps (individual spars)
6. Rib caps (individual ribs)

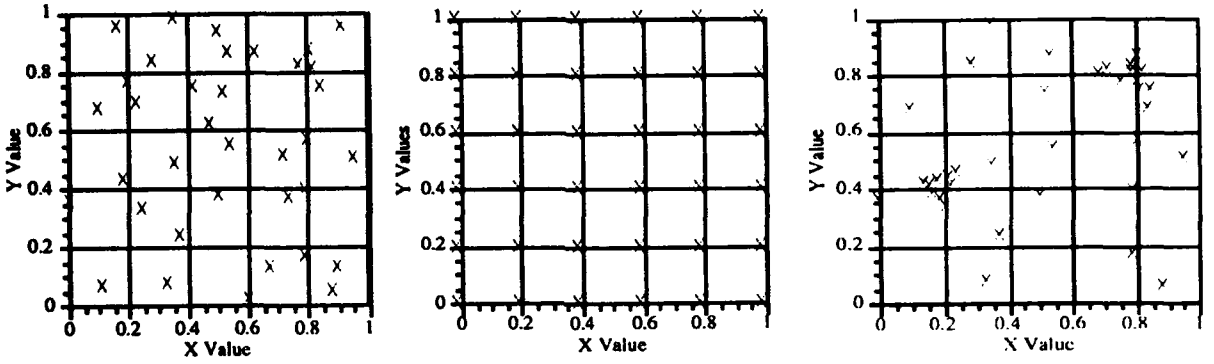
The automated assignment of material properties to all of the elements in the same element group was useful in reducing the information required in the model data preparation.

2.4 The Design Space

One remaining issue related to the finite element model development is the manner in which the individual design concepts were distributed within the structural design space. Depending upon the type of design variables being considered, a relatively large number of designs could be required to accurately define the design space. For all the design studies performed here, it was necessary to distribute the designs throughout the design space under consideration. For the simple five bar truss discussed in Section 1 and shown in Figure 1, a 61x61 grid of fully-stressed designs were evenly distributed throughout the design space.

Three different approaches were used to modify the finite element models according to the needs of the respective design study. The first approach involved assigning a uniformly distributed random value to each design variable. This approach was selected to avoid a user bias in the selection of the design variables. The second approach involved an evenly distributed set of values for the design variables in the allowable region of the design space. This approach helps to minimize the possibility that regions might have very sparse sets of designs, leading to a possibility of poor resolution of the design space in these regions. This second approach was also rather simple to implement and was computationally straightforward. The final approach involved the recursive learning procedure, in which an initial set of training data was generated based either on a random set or an evenly distributed set of design variables. Then, all further information on the design space were obtained based on the neural network predictions for promising regions of the design space. Figure 7 shows what these three distribution schemes might look like for a two-dimensional design space problem.

Figure 7a shows a random distribution of 36 designs based upon the independent selection of two continuous design variables, X and Y. As expected, the designs are relatively well dispersed. Figure 7b shows an evenly distributed set of 36 designs, while Figure 7c shows the type of distribution to be expected from the recursive learning procedure. There are two distinct regions of clustered designs shown in Figure 7c. These



a. b. c.

Figure 7. Distribution of Candidate Designs in a Two Dimensional Space

regions may be defined as regions of "desirable" design characteristics based upon a less refined description of the design space, and additional designs have been recursively added to these regions to better refine these portions of the design space.

Because the desired design space representations to be obtained from the studies in this report require design information from multiple finite element models, the development of a suitable procedure for the rapid development of these models was developed. The techniques used to select the element gages, or to perform "point" designs on each individual finite element concept will be described in the next section.

3. MULTIDISCIPLINARY DESIGN INFORMATION

This section deals with the methods used to develop the design information necessary to conduct the preliminary structural design studies. An overview of the analysis capabilities of the two programs from which all the design information was subsequently obtained, SWIFTOS and ASTROS, was presented in the previous section. Each of these programs also has the capability to determine optimum designs, in a least weight sense. When used simply for analysis, these finite element-based programs were only able to estimate the design based upon the sizes of each structural element as specified by the user. When the optimization capabilities of the program were used, each was able to determine the least weight structure based upon various design criteria. The basic purpose of this research was to develop the capabilities to select the "best" of the least weight designs as developed using these MDO procedures.

SWIFTOS used an optimality criteria approach to least weight design, while ASTROS used a math-programming procedure to obtain least weight designs. (Note that ASTROS has the capability of using optimality criteria like SWIFTOS but the latter was used for the reasons discussed earlier.) The benefits and limitations of both procedures, as well as the sequence of steps that must be performed to obtain the final designs from these programs, are outlined below.

3.1 Structural Optimization

As discussed, the structural design problem involves the selection of an appropriate set of design variables to achieve required performance at an acceptable "cost." One could feasibly pose a problem in a fashion in which all of the design variables, i.e. basic configuration, materials, and component sizes, were considered simultaneously and an iterative procedure developed to satisfy the constraints and to minimize the cost. In the approach to preliminary structural design used in this study, a two step approach was taken. This required the separation of the design variables into two classes. The first class of variables were used to parameterize the basic configuration geometry and material types, the second class was related to the size of each structural component. The design studies were conducted by selecting a finite subset of all possible designs as defined by the first class and then using the state-of-the-art MDO optimization techniques described below to determine the values for the second class for each specific design to achieve an optimum in a least weight sense. The design space defined by the first class of design variables and then spanned by each of the optimum designs was parameterized using the neural networks and subsequently evaluated to determine the "best" of the optimum designs.

Subsequent sections will discuss this approach and detail the application of the neural networks to this problem. This section provides an overview of the methods used to quantify the second class of design variables. Once the structure has been idealized and a finite element model constructed, an initial set of element "size" design variables were defined. For a given finite element model optimized using SWIFTOS or ASTROS only the proportion variables were considered; these were the membrane thicknesses and rod cross-sectional areas for those elements which were considered as variables in the design process.

These design variables were changed using a variety of iterative procedures that seek an optimum design in a least weight sense. These N design variables used to define the sizes of various elements within the structure can be written as an Nx1 vector **D**. A performance index M that is a single valued function of **D** must be determined and is used to measure the merit of competing designs. Because of the possibility of changing its sign or inverting, M can always be chosen so that the goal is

$$M(\mathbf{D}) \rightarrow \text{minimum} \tag{3.1}$$

The search must be carried out in an N-dimensional space that has Q constraints. In a general class of structural synthesis problems, these constraints can be formulated as Q functional inequalities

$$g_q(\mathbf{D}) \leq 0,\ q= 1,2,3.....Q \tag{3.2}$$

Points in the design space may be compared by determining the merit function at these points; a lower merit function indicates a "better" design. This comparison is only valid if both points represent acceptable designs. For the structural design problem a design is considered acceptable if no yielding, buckling, or other failure condition occurs (failure constraints). Also, the design variables cannot violate their established boundaries (e.g., max. and min. gages -- geometric constraints). The constraint set is almost certain to contain some complicated and perhaps implicit functions of D, demanding considerable machine computation for their evaluation. There may be constraints on deflections at specified locations in the structure, bounds on natural vibration frequencies, requirements about the avoidance of such aeroelastic instabilities such as flutter or divergence, etc. For simplicity, these performance constraints will be excluded from the following discussions though they were considered in a number of the design problems discussed later in the report.

There are two main approaches available for optimizing the structure, these are based upon either analytic techniques or numerical methods. An analytical technique relies on the algebraic combination of Equations (3.1) and (3.2), where the inequalities of (3.2) are usually converted to equalities. The advantage of the analytical approach lies in the parametric evaluation of M_{opt} as a function of the design variables. This parametric capability permits ready evaluation for the most efficient determination of the design variables. The disadvantage is the difficulty of formulation as the problem complexity increases (i.e., an increase in the number of design variables and constraints) as well as the inability to express the constraints as manageable algebraic functions.

Numerical approaches can be subdivided into direct and indirect approaches. The direct approach makes an intelligent and systematic search among the design variables to locate the minimum value of its objective function. This method is general, but is computationally expensive if large numbers of variables are involved.[4] Linear and non-linear math-programming techniques are the most prominent in this group of numerical techniques. Math-programming techniques can be characterized as search techniques which progress toward an optimum based on information available from the current design.

The indirect approach uses optimality criteria that are usually developed using an "intuitive" approach and are "expected" to produce the desired result. Fully-stressed design (FSD), as one example, is usually equivalent to minimum weight in a structure designed for strength[27]. Optimality criteria methods are those approaches in which an *apriori* statement of a condition or set of conditions is made along with an algorithm to alter the design variables in order to achieve that condition. When that condition is achieved (such as each element within the structure is stressed to its constraint value) then it is assumed that the desired minimum weight is achieved. Such procedures are fast. They can handle large numbers of design variables; thus, they are practical for the design of large structures.

While the direct numerical search procedures of nonlinear programming provide a correct approach for both strength and stiffness design, early attempts proved that they were somewhat inefficient for the design of large practical structures. FSD procedures are different from math-programming in that they attempt to reach a global minimum of the

merit function by forcing each design variable to a constraint surface, though this does not guarantee a global minimum. FSD is typically efficient and often yields near optimal solutions. Notable limitations in fully-stressing were established when, during the late fifties, it was shown that nonlinear programming[28,29] was the correct framework for the structural synthesis problem, and that FSD was generally incorrect for indeterminate structures. The fact that a true least weight structure is not generated for an FSD becomes of lesser importance when compared with the ease with which the improved design is reached[30]. Designs produced using FSD are often superior to those produced by alternate methods, notably in cases where rigorous optimization might be impractical, such as at the preliminary design stage.

For aircraft structures in which the design is governed exclusively by strength requirements, the fully-stressed design method offers an economical and reliable approach to optimizing structures having large numbers of design variables[31]. Although FSD methods cannot be proven to yield absolute minimum weight material distributions, their application has yielded practical and efficient airframe designs[32]. The ease of implementation and application of FSD certainly offsets the lack of a precise determination of the minimum weight design.

The current trend in finite element-based design optimization deals with multidisciplinary approaches to design, of which ASTROS is an example. No longer is analysis the only domain in which finite element methods can make significant contributions; they are now being accepted as important design tools when linked with optimization methods. ASTROS allows for the design of a finite element model with constraints on such things as displacement, natural frequency, and a range of aeroelastic performance parameters. Many other design constraints are feasible. The motivation for design tools such as ASTROS is that improved designs can be obtained by considering multiple design constraints simultaneously, and reducing the amount of time required for a typical design cycle.

One last feature of each of the methods discussed above which should be noted is that, in each case (optimality criteria or math programming) as the iterative process proceeds, information gained from the analysis of previous designs is always eliminated. The failure to exploit the information on each design as one proceeds through the "design space" prompted certain developments in this current effort.

3.1.1 Fully-Stressed Design - SWIFTOS

SWIFTOS was a procedure based on the application of a "fully-stressed" optimality criteria. Optimization, in this context, involved the determination of the "gage" of specified elements within the finite element model. These elements were designed to meet both stress and minimum gauge constraints, with the objective of minimizing the total weight (or mass) of the model. The following sequence was performed in the design cycle:

1. Specification of design variables
2. Specification of minimum gage for all design variables.
3. Analysis
4. Resizing algorithm
5. Convergence check.

The analysis provided information on the stress levels in all design elements. This iterative procedure involved a check on constraint violation(s), a design variable resizing, and a new finite element analysis of the model based on the newly sized elements. Once all design element constraints were satisfied (either by reaching a stress or minimum gage constraint), the process was concluded.

Two constraints were imposed on the design variables; maximum allowable stress and minimum gage. The minimum gage was defined during the design variable

specification. The rods were axial force members, with only tensile or compressive stress. The maximum stress constraint was defined by the material stress constraint (either tensile or compressive) provided. The quadrilateral, subparametric elements had a more involved stress constraint based on a von Mises stress criteria.

The design procedure altered the sizes of those elements and element groups that were specified as the design variables. In a manner similar to the element material designations, all element groups could be specified as design variables or individual groups, and individual group members were specified (i.e., spar 1, rib 3, etc.). Each member could consist of several elements; each of these elements was a unique design variable and was designed to a unique value of area or thickness. The following element groups could be specified as the design variables

1. Spar membranes (all, or individual spars)
2. Rib membranes (all, or individual ribs)
3. Skin membranes (all, or individual bays)
4. Connecting posts (all)
5. Spar caps (all, or individual spars)
6. Rib caps (all, or individual ribs)

As each design element group or member was specified, the minimum gage for that group or member was also required.

After the initial analysis, the element stresses were compared to the allowable stresses for all elements specified as design variables. Element thicknesses and areas were reduced or enlarged until either the applicable stress constraint was met or the minimum gage was reached.

For the rod elements, the axial stress was either tensile or compressive. The following relationships were used to modify the rod areas at each design iteration.

$$g_r = \left| \frac{\sigma_{xx}}{\sigma_{(T,C)}} \right| \qquad A_{NEW} = A^* = \left| \frac{F_r}{\sigma_{(T,C)}} \right|$$

$$\begin{aligned} &If\ g_r < 1.0 \quad A_{NEW} = A^* g_r \\ &(0.9 \le g_r \le 1.0) \quad A_{NEW} = A^* g_r^2 \end{aligned} \tag{3.3}$$

If $A_{New} < A_{Minimum}$ then $A_{New} = A_{Minimum}$

and F_r = rod force

There were multiple conditions that could be applied in determining the new rod area. For values of g_r less than 1.0 it was noted that an increased reduction rate in area decreased the required number of design iterations. The area relations for $g_r < 1.0$ reflected this.

The membrane elements had three stress components that were accounted for using the von Mises stress constraint. The following procedure was used to determine the membrane thicknesses at each iteration.

$$g = \sqrt{\left(\frac{\sigma_{xx}}{\sigma_{(T,C)}}\right)^2 + \left(\frac{\sigma_{yy}}{\sigma_{(T,C)}}\right)^2 + \left(\frac{\sigma_{xx}\,\sigma_{yy}}{\sigma_{(T,C)}^2}\right)^2 + \left(\frac{\tau_{xy}}{\sigma_{(S)}}\right)^2}$$

$$\begin{aligned} &If\ (g > 1.0)\quad t_{New} = g\,t_{original} \\ &If\ (g < 1.0)\quad t_{New} = g^2 t_{original} \\ &\ or\ (0.9 \le g \le 1.0)\ t_{New} = g^3 t_{original} \\ &If\ t_{New} < t_{Minimum} \rightarrow t_{New} = t_{Minimum} \end{aligned} \qquad (3.4)$$

It has been noted that a reduced number of iteration cycles were required when a greater reduction rate in thickness occurs (for g < 1.0). The thickness relations for g < 1.0 reflected this.

After each iteration the resultant element stresses were again compared to the allowable stress. Thicknesses and areas were modified, if necessary, until a convergence of the solution (i.e., all element stress or minimum gage constraints were satisfied) occured. The termination of the iterative process was based upon satisfying a prescribed convergence criteria. The solution was said to have converged when all element stresses were within $\pm$ x% of the applicable stress constraints, or at minimum gage. For the cases considered in this report, a $\pm$ 5% convergence condition was selected. This convergence criteria had a direct influence on the uncertainty in the resulting "optimum design" and had a bearing on the eventual quantitative description of the design space. The convergence criteria could be modified. A reduction in this value would directly translate into an increased number of design iterations. All specified design elements were redesigned at each iteration, and typically when all elements met the $\pm$ 5% condition the majority of the elements were designed to within 1% of their respective stress constraint. The selection of the 5% tolerance was based upon earlier experience gained with SWIFTOS and an attempt to expedite the computations.

One last consideration related to the use of the optimality criteria, where no assurance of either local or global miminum was made, was that of the "initial" design. Though the initial design was usually selected to be of uniform "gage" some results are provided which illustrate the influence of the starting values of the design variables on their final, "optimum" values.

3.1.2 Math Programming Optimization - ASTROS

ASTROS is a large, multidisciplinary design program. Math-programming was used for the majority of optimization problems, but fully-stressed design was also available. Design variables could represent rod areas, shear and membrane element thicknesses, bar areas, and concentrated mass values. Constraints could be placed on stress-strain, displacement, modal frequency, and aeroelastic effects (lift effectiveness, aileron effectiveness, divergence speed, flutter...). Multiple boundary conditions and multiple loadings could be included in the problem. The goals of such a program are to emphasize the interdisciplinary features of the design task, reduce design time, and provide an improved design.

One would most likely consider ASTROS to be applicable to preliminary structural design at the point where the basic geometry of the structure as well as the appropriate material types had been defined. At this point the more complex interaction of strength design and aeroelastic constraints can take place. It was the goal of the current effort to establish an approach whereby an MDO procedure could efficiently be used in the early stages of the design process to influence the selection of the basic structural configuration and material system. It is, of course, difficult to describe concisely a

program as large and involved as ASTROS in the limited space available in this report. Only those aspects of ASTROS pertinent to the current applications are described. The following comments are meant to be informative descriptions and not detailed theoretical developments.

The mathematical programming techniques used in ASTROS perform a search for the optimum based on currently available information, such as the gradients obtained for the design variables in a given design iteration. Math-programming techniques are general in application, typically being robust procedures; however math-programming is also computationally intensive. The practical limitations imposed by present computer resources allows for the consideration of 200-300 design variables at most for a given design problem. ASTROS uses an algorithm which was developed in order to exploit a combination of features from the feasible directions[33,34] and generalized reduced gradient[35] algorithms. The procedure performs a one-dimensional search based on a polynomial interpolation with bounds.

In ASTROS, a major efficiency gain can be realized by approximating quantities in redesign rather than computing them explicitly. This is termed the approximate design problem. In the approximate design problem an assumption is made that constraint gradients are invariant with respect to changes in the design variables. The quality of this assumption is enhanced by the use of inverse design variables ($X_i = 1/V_i$). The motivation for this is that strength constraints are inversely proportional to structural thickness. Due to this assumption limits are imposed on the changes in design variables during design to provide a more stable convergence in the optimization.

A number of element types (axial force, shear panel, quadrilateral membrane, and concentrated mass) were used as design variables. Other finite element types are available in ASTROS but were not used for the applications considered here. For the above finite elements, the mass and stiffness matrices are a linear function of the design variable.

A number of design constraints were also available in ASTROS. The applicable stress-strain constraint based upon a von Mises constraint was used for isotropic materials. A natural frequency constraint was imposed in a number of the applications that used ASTROS.

Gradient information was required by all math-programming design approaches. The gradient information provided the sensitivities of the objective function to each design variable as well as the constraint sensitivities to each of the design variables. The gradients of the objective function were considered invariant--this is a key to performing the approximate design problem. The gradients of the constraints were all computed analytically, leading to a number of "intricate" computations. There were many involved sensitivity analysis developments in ASTROS. The static analysis strength, natural frequency and flutter constraints were used in the current effort.

The final consideration related to the development of the design information using ASTROS is related to the manner in which the iterative design process was terminated. A specific design was considered to have converged when the following criteria for the approximate problemwas met:

$$|\Delta M / M| \leq 0.005 \qquad (3.5)$$

where M was the previous value of the objective function and ΔM was the change in the objective function. The 0.005 value was user definable, and it was noted that a reduction of this value resulted in a more stringent design termination criteria which produced designs that were more consistent in representing what appeared to be the true global least weight design. By making the termination criterion more stringent,the design was effectively forced to evolve several more iterations with an accompanying reduction in weight. The resulting designs can be thought to have a smaller "uncertainty" than designs

obtained with a larger termination parameter. The draw-back is that more iterations must be performed with a resulting increase in computational cost.

Once the above criterion was satisfied a check was then made to determine if all constraints were satisfied. The design was considered to have converged when

$$2.0 \cdot \mathrm{CTL} < g_{max} < 3.0 \cdot \mathrm{CTLMIN} \tag{3.6}$$

where

g_{max} – Maximum constraint value
CTL – Active constraint identifier (default = −.003)
CTLMIN – Violated constraint identifier (default = .0005)

where CTL and CTLMIN were user definable parameters. Though no detailed study of the influence of the design iteration convergence criteria was performed, it is important that one considers the uncertainty in design when evaluating the suitability of a particular design concept.

3.2 Point Design Detail vs. Design Space Definition

One final issue briefly addressed above is related to the distinction between defining an "optimum" point design and providing a quantitative description of the design space. The two design methods discussed above provide the means to define a "feasible" design given its basic geometry and materials. The suitability of the final design is dependent upon the accuracy of the definition of the constraints and the ability of the optimization algorithm to provide a realistic "minimum weight." The final design is defined in terms of the set of design variables which are related to the size of individual components, performance measures (flutter speed, deformation, etc.), and a merit function (weight). ASTROS can also provide information on sensitivities but these too are "local" properties of the final design. No quantitative information is provided as to the dependence of the design on a variety of important design variables.

This is a important limitation particularly in the early stages of the preliminary design process. It would be ideal if the detailed information developed as the result of the finite element analysis and subsequent numerical optimization could be used in a more effective fashion in the early stages of structural design. It is to this end that the efforts of this research were directed. This section has focused on those aspects of the design software employed for the design studies relevant to the applications that were considered. In the subsequent sections, the methods discussed herein are used to provide detailed design information and are the primary source of "knowledge" as one attempts to quantify the structural design space.

4. NEURAL NETWORKS

This section provides an introductory discussion of feed-forward, back propagation, neural networks. Features of the neural network program employed for this research, NETS[36], is also described and the back propagation training algorithm is discussed.

4.1 Introduction to Feed-Forward Neural Networks

Neural networks provide a promising approach that may be used to interpolate or extrapolate from a set of design information. Neural networks are essentially simple mathematical models that attempt to mimic the neural structure and computational elements of biological neural systems (such as the human brain). It is recognized that the human brain performs many types of processes much faster than even the most involved artificial intelligence system running on a supercomputer, leading to an interest in computational modeling of these types of biological systems. The brain has other features that make this investigation desirable:

- It is robust and fault tolerant. Nerve cells in the brain die every day without affecting its performance significantly.
- It is flexible. It can easily adjust to a new environment by "learning" -- it does not have to be programmed in Pascal, FORTRAN, C, etc.
- It can deal with information that is fuzzy, probabilistic, noisy, or inconsistent.
- It is highly parallel.
- It is small, compact, and dissipates very little power.

The history of neural modeling dates back to the paper of McCulloch and Pitts[5], in which they derived theorems related to models of neuronal systems based on what was known about biological structures in the early 1940's. Around 1960 there was a resurgence of neural network activity centered around the group of Rosenblatt[7] focusing on the problem of how to find appropriate weights (w_{ij}'s) for particular computing tasks. The weights are the medium by which information is stored in the network and provide the means to perform a mapping of inputs to outputs. Rosenblatt concentrated on perceptrons (simple, two-layer, feed-forward networks). In 1962, Rosenblatt proved the convergence of a learning algorithm, a way to change the weights iteratively so that the desired information could be stored within the network. Unfortunately, in 1969 Minsky and Papert[37] pointed out the limitations of Rosenblatt's perceptron work and the inability of the single layer perceptrons to perform some elementary computations such as the exclusive-or operation (XOR). It wasn't until 1974 that Werbos[38]discovered the back-propagation algorithm that could be used to determine the weights in successive layers of multilayer perceptrons. This was independently rediscovered around 1985 by Hart, Hinton, and Williams[39], and by Parker[40]. Development of this learning algorithm was central to the resurgence of interest in neural networks. The back-propagation algorithm has gained much favor in the neural network community and much activity is currently centered on back-propagation and its extensions. There are many important implications to training a network to perform a computation. Rather than specifying every detail of a calculation only a representative set of training examples has to be constructed. This means that problems where appropriate rules are difficult to determine in advance can be treated. A recent count indicated that there were 26 various types of neural networks commonly in use.[41]

A single neural network may be composed of thousands of neurons. This is presently the level at which networks can be simulated with either software or hardware. In contrast the human brain is composed of more than 10^{11} neurons (with more than 1000 times as many interconnections). The typical activation time of biological neurons is a few milliseconds, which is about a million times slower than a neuron implemented in computer hardware. Nevertheless, the brain can perform very fast processing for tasks like vision, motor control, and decisions on the basis of incomplete or noisy data - tasks that are far beyond the capabilities of a computer. There is a sharp contrast between the type of information processing performed by the neural network and the conventional approach. The neural network has many processors, each executing a simple program, instead of the conventional situation where one or at most a few processors execute very complicated programs. And in contrast to the robustness of a neural network, an ordinary sequential computation may easily be ruined by a single bit error.

Although the original research into neural networks was aimed more towards an understanding of the physiological aspects of human thought, much current work deals with determining applications in science and engineering for artificial neural networks (which exhibit many of the characteristics of their biological counterparts). The recent interest in artificial neural networks has brought about an entire vocabulary for describing these networks. Much of this terminology has its roots, understandably, in biology. This terminology is now discussed.

Figure 8 shows a simple neural network configuration. In this example there is a "two-element" input layer, a hidden layer of eight neurons, and an output layer with a single output neuron. This neural network can be described as a parallel system of nine processors termed neurons (the shaded "circles" are not neurons, but simply inputs to the network). Each of the neurons has a very simple program. It computes a weighted sum of the input data from the preceding layer of neurons (or directly from the input layer) and then outputs a single number which is a nonlinear function of the weighted sum. This output is then sent to other neurons (in following layers), which are performing the same type of calculation. The other neurons will be using other weights -- those weights applicable to their inputs. These various weights can be thought of as the medium in which information is stored within the network. The number of nodes in the input and output layers is defined by the problem to be solved. The hidden layers aid in performing the mapping between an input and its output.

Consider Figure 9 which represents a single neuron. This is the basic building block of an artificial neural network. There are two major items to be noted in the diagram. The first is the processing element which is referred to as a neuron. The neuron's function is basically analogous to its biological counterpart. It uses a set of incoming connection values to calculate its output through an activation function. The incoming connections on the left of the neuron connect the outputs of previous neurons to the current neuron. The connections are analogous to the synapses of biological neurons. Each network neuron may have many incoming and outgoing connections. These connections are made with other neurons in the network. Each neuron produces only a single output value. This output is seen as an input to the next neuron layer. Each connection has an associated weight. These weights effectively modify the signal that passes along this connection. The training, or learning process involves the modification of these connection weights. Training is an iterative process which involves the processing of a set of data by the network. The weight values are modified until the network represents all of the data (within the training set) to some acceptable level of error.

One of the most common functions used to represent the computational characteristic of a single neuron is the sigmoid function:

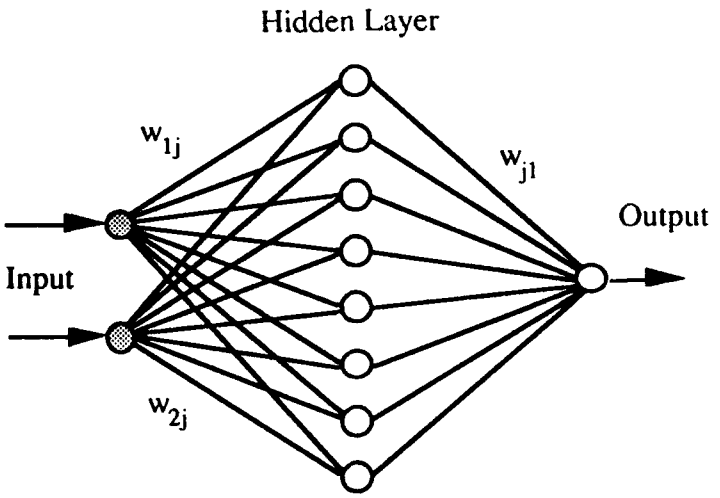


Figure 8. A Simple Neural Network

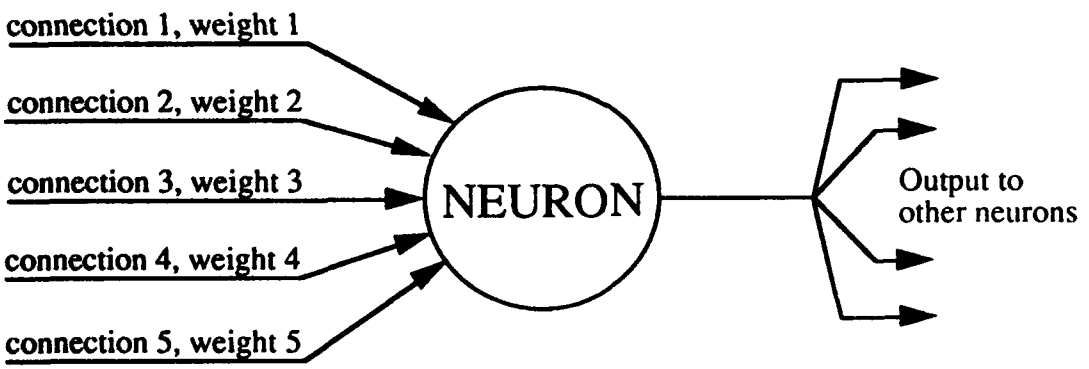


Figure 9. A Single Neuron

$$y_j = \frac{1}{1 + e^{-\beta}} \tag{4.1}$$

where

$$\beta = \sum_{i=1}^{N} (w_{ij}X_{ij} - \theta_j). \tag{4.2}$$

The sigmoid function was used for the neurons in this work. The w_{ij} coefficients are the weights. The value y_j is the neuron output and the X_{ij} values are the inputs. The θ_j coefficient is termed the bias, and is also adjusted during network training. The output of the sigmoid function is shown in Figure 10 as a function of the parameter β. Note that the output of the sigmoid function is in the range of 0.0-1.0. This has important implications in the formulation of training data, in particular the scaling of the training data to occur within this range. This will be discuss in Section 4.2.5.

The process used when feed-forward, back-propagation networks are employed to perform a mapping of a given set of information can be considered as a sequence of steps. The first step is the designation of the input and desired output from the network in the form of input/output vectors or pairs (IOP). The number of neurons and their connections must be established. Finally the parameters used to describe each neuron must be determined. The determination of these parameters can be expressed as an optimization problem. The minimization of error between the network output and the desired network output is termed *network training*, and will be discussed in the following section.

4.2 NETS

NETS[36] was the neural network simulator program used for this research. It was developed in the Artificial Intelligence Section of NASA's Johnson Space Center. The stated purpose of NETS is:

> "(1) to provide a somewhat flexible system for manipulating a variety of neural network configurations using the generalized delta back propagation learning method and (2) to provide the general user community a means of learning about neural network technology without the need for specialized hardware."[36]

In the case of this research, NETS satisfied both of these purposes very well.

4.2.1 Back-propagation Training

Once a network configuration has been decided upon and a set of training data generated, the process of network training begins. Training involves the modification of the network coefficients (the weight and bias values) to obtain the desired mapping of input to output which is based on the set of input/output pairs used as the training data. This effectively involves a comparison of the desired and actual network mapping, and a lowering of the level of error between the two. A user supplied error constraint specifies the termination criterion for training. When the specified allowable error level is reached, training stops.

Back propagation uses a gradient descent technique to define the weights in the network. The error between the current network output and the desired output is squared to obtain a function that can be used to perform the gradient descent for weight modification. The back-propagation description was developed by Hertz et al[42]. This iterative algorithm provides a mechanism for changing the weights in any feed-forward

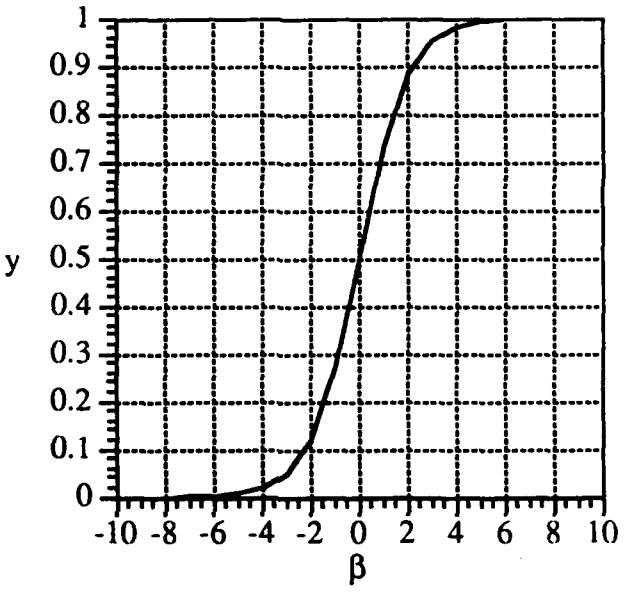


Figure 10. Sigmoid Activation Function

network to learn a training set of input-output pairs. The weights can be binary (0/1) or continuous valued. Continuous valued weights were used in this application. The addition of a momentum term to the weight update equation can significantly improve the characteristics of the gradient descent and improve the convergence rate. Additional details of the back-propogation algorithm can be found in many texts on neural networks or in Reference 8.

4.2.2 Learning Rate

The learning rate is simply a multiplier that is used as part of the weight changing procedure of back propagation. The error thus relates not only the current accuracy of the network, but the rate of change to be made to the weights during training. The larger the error, the larger the change. Since back propagation only approximates a true gradient descent, the learning rate is used to scale the degree of change made to the network so that as close to a true gradient descent as possible is achieved. A large learning rate will provide faster learning, but the error level will fluctuate greatly (as the mean error level shows a decrease). For large learning rates, the network may be unable to achieve the final desired level of error.

In order to improve training efficiency NETS incorporates a scaling factor for the learning rate. Recent work[43] has shown that training time can be reduced if the learning rate is adjusted during the learning process. This is particularly true when error levels are either very high or very low (as is typical at the beginning or ending of training). The scaling factor is shown in Figure 11 as adapted from Reference 36. There is a sharp increase in the scaling factor at both error extremes, but when the error is in the range of 0.5 the scaling factor is 1.0. The end result was that there was an increase in the learning rate when the error was either large or small.

4.2.3 Delta Weight Constant

Rumelhart and McClelland[44] describe a term that can be added to the weight change calculation to make the training process more efficient. This factor is also known as the momentum. The problem is that the gradient descent can be very slow and can oscillate widely if the learning rate is small. The momentum is based on the last change made to each weight. When a weight is changed, a part of the last weight change is also added. Then the effective learning rate can be made larger without divergent oscillations occurring. This scheme is implemented by giving a contribution from the previous training iteration to each weight change. The momentum parameter α must be between 0 and 1.0. A value of 0.9 was chosen in this work.

4.2.4 Neuron Bias

The bias coefficient does just that; it provides an offset or "bias." The bias values allow for the representation of more involved and complex relationships in the data to be mapped. There is a single bias value for each neuron. Figure 12 shows the effect that various bias values have on the neuron activation function. Notice that the bias value effectively "shifts" the sigmoid function output along the horizontal axis. The neuron bias is also often referred to as the activation threshold. The bias is determined as part of the network training.

4.2.5 Scaling of Input/Output Pairs

Until the training process has been completed, the neural network has no information stored within it. Training data in the form of sets of input/output pairs are

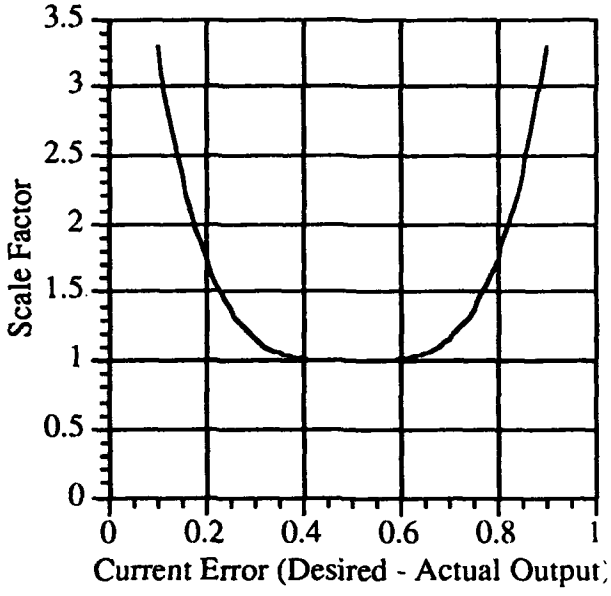


Figure 11. Scale Factor

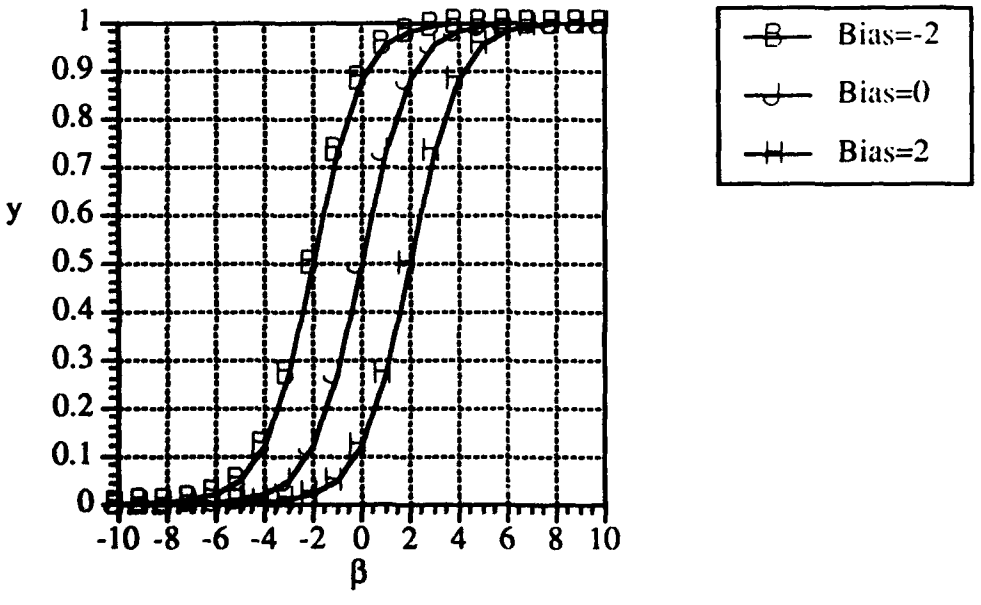


Figure 12. Effect of Bias on the Sigmoid Function

used. These input/output vector pairs are typically scaled to the range somewhat greater than 0 and less than 1. This is due to the fact that the values of 0 and 1 are difficult to obtain (the value of the derivative used in the back-propagation procedure being 0 at these points). For most of the problems considered in this work the training data was scaled in the range from 0.1 to 0.9.

If the design information to be mapped was continuous in nature, a linear scaling was employed (in the range of 0.1 to 0.9) based on the minimum and maximum values in the training data for the network output(s). As an example of the scaling procedure, consider the case where the set of design information to be mapped by the neural network has a maximum value of 250.0 lb and a minimum value of 150.0 lb. The 250.0 lb value is scaled to 0.9 and the 150.0 lb value is scaled to 0.1 and all intermediate values are scaled using a simple linear scaling based on the maximum and minimum values. This scaling procedure was performed for all output data (if multiple network outputs were used). The same type of scaling was also performed on all input data.

For the case of discrete valued design information, a different type of scaling on the training data is performed. As for the input data, the values are set to either 0.1 or 0.9 relative to the state of the discrete design variable. The discrete design variable applications to be presented involve material selection, thus each design variable can be one of several possible discrete values. To account for this, each design variable is assigned a set of network inputs corresponding to the number of discrete states. If there are four materials being considered for an element of the structure (four discrete states), then four inputs to the network are associated with this design variable. For any given training pair only one of these inputs will have a value of 0.9 -- that input corresponding to the active discrete state (or material selection). The other inputs are set to values of 0.1. If, in a material design problem, rod 1 is composed of material 3 (of four possible choices), then the four inputs to the network associated with this design variable are (0.1, 0.1, 0.9, 0.1). A variation on this type of discrete variable scaling was used in the composite wing skin application discussed later in the report. This is presented in more detail at that time.

The development of the information required to provide the training data and the selection of the format and scaling of this information has a large impact on the ability of the neural network to effectively represent the relationship between the "input" parameters or design variables and the "output" or measures of merit or performance.

5. APPLICATION OF NEURAL NETWORKS TO PRELIMINARY STRUCTURAL DESIGN

This section presents the methods which were developed to integrate the MDO procedures with the capabilities of the artificial neural networks. The concept of recursive learning is also introduced. The two optimization approaches taken to extract design information from the neural networks - successive quadratic programming method and simulated annealing, are then discussed. Finally, a specialized application is presented in which the characteristic of "survivability" of the design is considered.

5.1 Mapping the Design Space

Once trained, the neural network weights and bias values were saved. The network could then be used as a static entity, typically as a functional subroutine to another program to either display the variation of output parameters as a function of the input parameters or to identify regions in the design space of particular interest. No modifications to the weights were made beyond the termination of training. In the case of recursive learning, which will be discussed later in the report, the weight and bias values were stored as static entities at each iteration after having been modified as the result of the addition of new training data.

To illustrate a design space representation using neural networks, consider the example of the five bar truss of Figure 13, which was also discussed in Section 1. The truss is composed of five axial force rods, constrained at nodes 1 and 4, with the applied loads shown. Consider as a possible design problem that one would allow the node 2 location to range over the 6 by 6 inch region shown. It would be desirable to quantify the design space defined by the weight variation for all optimal designs within this region. This could be used to determine if some portion of this design space has weight characteristics that are "better" than other regions. Determining the trends in weight may be as important as determining the single best design for this structure. This requires the determination of a design space representation. Such a representation was obtained by using an automated FSD procedure (mesh generation and design combined) with constraints on tensile, compressive and buckling stress to generate a set of points in the design space as described earlier. These points represent optimal designs for unique geometries. A 61 x 61 grid of evenly spaced designs within the region were obtained (3721 designs). Based on this data, the contour plot of Figure 14a was generated.

Figure 14b shows a contour plot obtained from data developed using a neural network. The network was trained with 121 FSD truss designs distributed over the allowable design space (an evenly distributed 11 x 11 grid). Once trained, the network calculated the FSD truss weight based on a 61 x 61 grid of (x,y) coordinates for node 2. The network output for weight was then contoured as was done in Figure 14a for a similar grid composed solely of FSD designs. It is easily seen that the neural network has been able to distinguish the major characteristic features of the design space using more than an order of magnitude less information than that required to produce Figure 14a. It is this ability of the neural networks to quantify, store and then provide a means for easy recall that makes them particularly attractive for processing preliminary design information.

5.2 Recursive Training of Neural Networks

The feed-forward networks employed in this study require training data to form a design space representation. Because FEM and MDO procedures are typically computationally expensive, attempting to minimize the amount of training data required for an effective neural network representation must be considered. Training data generation is the most computationally intensive part of the design procedure. To address

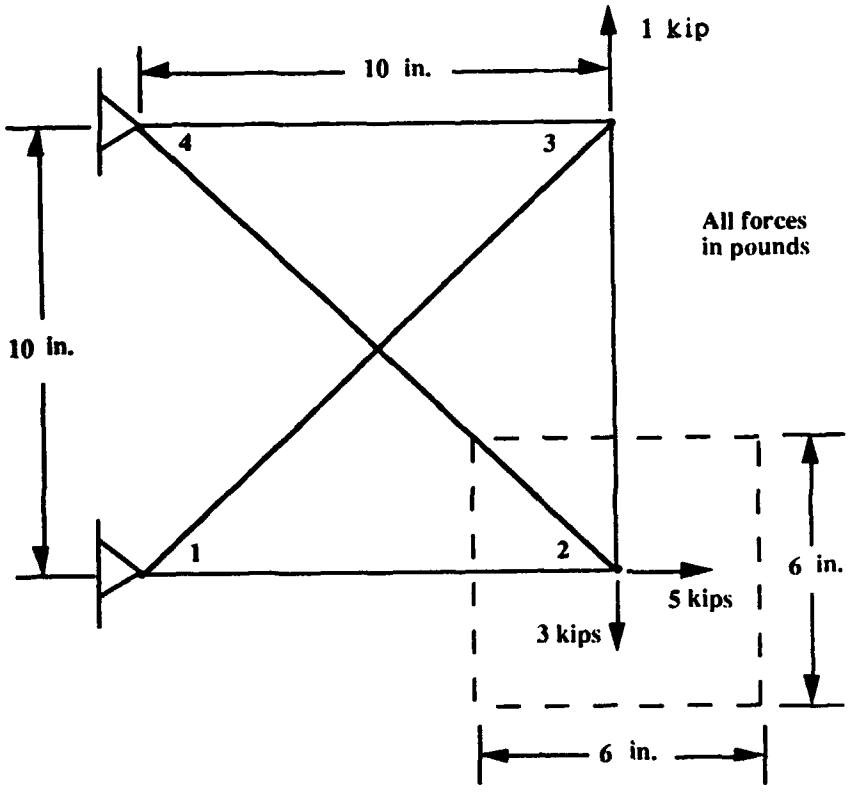
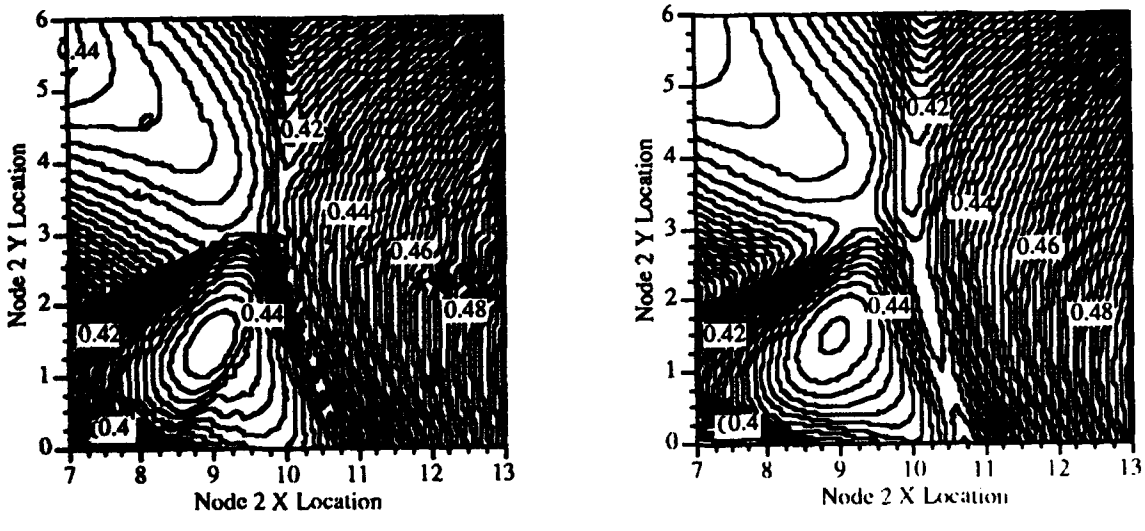


Figure 13. Five Bar Truss Schematic



a. Actual

b. Neural Network

Figure 14. Weight Contours for Five Bar Truss

this issue an automated finite element mesh generator and recursive learning procedure were developed. The mesh generator was an extension of the original version of the SWIFTOS preprocessor XPUT.

Given a small set of designs (a function of the design problem complexity), a neural network was trained to approximate the design space of a given structural design problem. Once trained, the neural network was interfaced with a math-programming optimization program which was used to predict regions of the design space that were added to the training set because of the desirable design characteristics associated with those regions. The preprocessor generated new finite element models (and associated MDO input data) appropriately based on the design space region (i.e., values of the design variables). The sizes of the components of the specific models were designed to provide additional "optimum" designs, and were then incorporated into the existing training set. The neural network was then retrained with the new design information to better represent these regions of the design space. The procedure was recursive in that the sequence was repeated until it had been determined that the neural network representation of the design space was refined enough for the problem under consideration.

A flowchart of the recursive training procedure is shown in Figure 15. The entire procedure was implemented through a cross-compilation of FORTRAN and C code, and the program was a stand-alone tool requiring minimal interaction with the user. The neural network was incorporated into the analysis procedure so that the set of training data was automatically updated as new information was generated. Training occurred sequentially with the generation of data.

5.3 Design Information from Neural Networks

This section provides background on the procedures that were employed to search the neural network design space representations so that improved designs could be determined. Two procedures were employed. The first was a successive quadratic programming method; this procedure was used to determine optimal configurational designs for continuous valued design variables. The second optimization approach employed was the simulated annealing algorithm. The simulated annealing algorithm has a recognized utility for combinatorial optimization problems. In this study simulated annealing was employed for the material system design problems where the design variables were discrete valued.

5.3.1 Systematic Search - Nonlinear Programming

The implementation of the successive quadratic programming method to solve a general nonlinear programming problem was based on a method developed by Schittkowski[45]. The implementation of this program was taken from the subroutine NCONF of the IMSL[46]library of mathematical subroutines. The method, based on the iterative formulation and solution of quadratic programming subproblems, obtained subproblems by using a quadratic approximation of the Lagrangian function and by linearizing the constraints. The augmented Lagrange function was used as the merit function. A finite difference method was used to estimate the necessary gradients in the calculation. For the neural network implementation, a neural network output was treated as the objective function, typically weight, and any other network output could be used as a constraint on the optimization. For the cases considered, only upper and lower bound constraints were considered on the optimization problems.

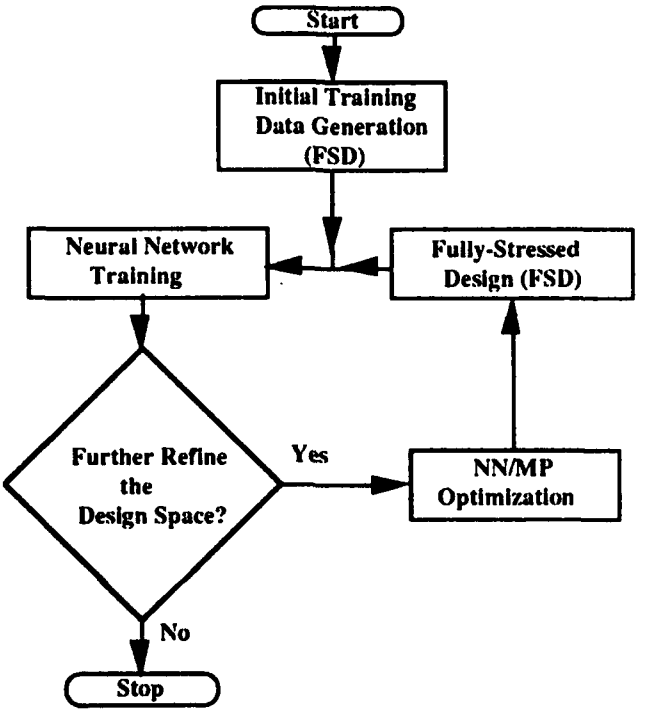


Figure 15. Flowchart of the Recursive Learning Procedure

5.3.2 Simulated Annealing - Discrete Design Variables

Conventional optimization methodologies, which are typically based on sensitivity analysis and gradient-based search techniques, are not suitable for the discrete, or combinatorial, optimization problem. One example of this would be determining the material system for a candidate structural concept (from a discrete number of material possibilities) to obtain a least weight material distribution satisfying all applicable constraints on the design. As the number of combinations increases, the possibility of an exhaustive search becomes prohibitively expensive and eventually impossible. Random search techniques, which evaluate a limited set of possible combinations are useful for only smaller combinatorial problems where the possibility of an exhaustive search is reasonable. As the combinatorial size grows, the random search becomes less valuable and inefficient. It is understandable then that optimization problems dealing with discrete design variables, while often mathematically well defined, are typically difficult to solve in practice. New, and efficient techniques to allow for the consideration of discrete design variables are thus necessary.

Simulated annealing (SA), which is based on a strong analogy to the annealing process in metallurgy, has been used effectively for combinatorial optimization problems[47]. SA typically requires large numbers of design iterations to obtain near optimal solutions. Rather than using costly FEM analysis for the simulated annealing procedure, neural networks were used as an effective medium for the approximation of the design space of a given structural concept and set of material systems. Neural networks provided the computational efficiency to perform these SA optimizations in an expedient manner.

5.3.2.1 Basic Algorithm

Kirkpatric, Gelatt, and Vecchi[48](and independently Cerny[49]) provided the foundation for applying annealing to combinatorial optimization problems. The basis of simulated annealing can be traced to Metropolis[50]. Simulated annealing has been used effectively for such engineering applications as the optimal placement of modules on an integrated circuit[51-53] and image processing[54-56].

Most engineers are familiar with the annealing process as described in metallurgy; a heat treatment in which a metal is subjected to high temperatures so as to soften the material, and then recrystallized upon cooling. The process involves the heating, and then a slow, controlled temperature decrease until the atoms arrange themselves in the ground state of the solid. While at high temperature, the atoms are arranged randomly, but when at the ground state (after slow cooling) the atoms are arranged in highly structured lattices where the energy of the system is minimized. To ensure that the energy state is a minimum (or near minimum), the initial temperature (randomness) must be high and the cooling rate must be sufficiently slow.

The following is a summary of the simulated annealing description by Aarts and Korst[57]. The simulated annealing algorithm is based on Monte Carlo techniques and generates a sequence of states of the design vector. Given a current design vector, **i**, with associated objective function E_i, then a subsequent design vector **j** is generated by applying a perturbation mechanism that transforms the current vector into the next by a small distortion. The objective function of this vector is E_j. If the objective function difference E_j-E_i is less than or equal to 0, the design vector **j** is accepted as the new design vector. If the objective function difference is greater than 0, then the vector **j** is accepted with a certain probability. This acceptance rule is termed the *Metropolis criterion*. It is developed using the following expressions.

$$M = e^{\Gamma} \tag{5.1}$$

where

$$\Gamma = \frac{(E_i - E_j)}{k_B T} \tag{5.2}$$

where T denotes the temperature of the "heat bath" and k_B is a physical constant known as the *Boltzmann constant*. The above exponential is evaluated, and the resulting value is compared to a random number. The random number is selected from a uniform distribution of random numbers in the internal (0,1). A value of the parameter M greater than the random number results in an acceptance of the new design vector. The algorithm associated with this acceptance rule is known as the *Metropolis algorithm*.

The Metropolis algorithm can be applied to generate a sequence of solutions of a combinatorial optimization problem. The following analogies are made:

- Possible solutions in a combinatorial optimization problem are equivalent to states of a system.
- The objective function of a solution, f(i), is equivalent to the energy of a state, E_i.

These conditions are typically implemented by,

$$P_c\{accept\ j\} = \begin{cases} 1 & \text{if } f(j) \le f(i) \\ e^{\left(\frac{f(i)-f(j)}{c}\right)} & \text{if } f(j) > f(i) \end{cases} \tag{5.3}$$

where **i** and **j** are two solutions of a combinatorial optimization problem with objective functions f(i) and f(j), respectively. The *control parameter*, represented by "c" in Eq. 5.3, will be discussed shortly. The Metropolis algorithm involves nothing more than a transition from state **i** to state **j** consisting of the following two steps:

(1) application of the generation mechanism,
(2) application of the acceptance criterion.

As shown in Eq. 5.3, if $f(j) \le f(i)$, design vector **j** is accepted as the new design vector. However, if $f(j) > f(i)$, vector **j** may still be accepted based on a comparison between the Metropolis criterion and a uniformly distributed random number on the interval (0,1). If the Metropolis criterion provides a value greater than the random number, design vector **j** is accepted as the new design vector. If the Metropolis criterion value is less than the random number, the perturbation made to design vector **i** is eliminated, and a new perturbation to design vector **i** is applied.

For a given application of the simulated annealing algorithm, a generation mechanism appropriate for the optimization problem, as well as a control parameter schedule, must be determined. These issues will be discussed in the following Sections, and this is followed by a simple SA demonstration using the five bar truss material design example discussed in the Introduction.

5.3.2.2 Generation Mechanism

As mentioned previously, the generation mechanism defines how the discrete design vector was perturbed between consecutive iterations of the simulated annealing algorithm. There were two concerns when selecting the generation mechanism:

(1) representing any physical behavior that might be characteristic of the discrete phenomena being modeled, and
(2) creating an efficient mechanism from the standpoint of the simulated annealing algorithm's execution.

In the latter case, it would be advantageous to reduce the number of simulated annealing iterations to as few as possible, while still obtaining an objective function value close to the global minimum. The modifications and re-evaluations of the design vector were the most expensive portion of the simulated annealing algorithm. It was thus desirable to make this process as efficient as possible. The generation mechanism is usually chosen so that the new solutions can be obtained from current ones by simple rearrangements that can be computed rapidly, for instance permutations, swapping, and inversions. It is preferred that calculation of the objective function be done incrementally, taking into account only the differences resulting from the local rearrangements. The generation mechanism was also likely to be "time-dependent," so that the mechanism would change as the number of iterations increased (the mechanism will cause "smaller" modifications). This was meant to better represent the cooling process, and in effect caused the optimization to slowly settle into the global minimum. For the case of neural networks, the objective function computation was quite efficient, and the need for an involved generation mechanism was greatly reduced.

The simulated annealing implementations for this work employed only a simple random generation mechanism. The mechanism involved the selection of a random subset of the design variables. These design variables were then perturbed in a random fashion. A new value was selected at random for each of these design variables. The objective function was then recalculated. For the ten bar truss material design example (presented in the following section) a single rod's material selection was modified at each SA iteration. For the ACOSS II example, five rods had their material selections change at each SA iteration. These changes involved the selection of one or five rods at random (from the entire set of rods for the particular example), and then the assignment of a new material for the rod(s). The newly assigned material was also selected at random from the four material possibilities.

5.3.2.3 Control Parameter

The control parameter, denoted by "c" in Eq. 5.3 and analogous to the product of k_BT in Eq. 5.2, effectively modifies the acceptance criteria. Typically, careful planning of the control parameter schedule was necessary to minimize the number of simulated annealing algorithm iterations. The control parameter had some initial value at the start of the annealing process, and was then reduced in magnitude as the annealing process proceeded. In practice the control parameter could be a function of the current generation mechanism change (accounting for a local change in the design variable vector) as well as the SA iteration number. This was done again for efficiency.

Because of the efficiency of the neural network representation, a more involved control factor schedule was not deemed necessary for this work. The control factor schedule employed for all the simulated annealing optimization applications is shown in Figure 16.

The control parameter schedule (which was a function of the iteration number N) was defined by the equations:

$$c = 5.00\text{E-}03 \qquad (N < 100) \tag{5.4}$$

$$c = A*N^B \qquad (100 < N < 5000) \tag{5.5}$$

where A=200.0 and B=-2.2

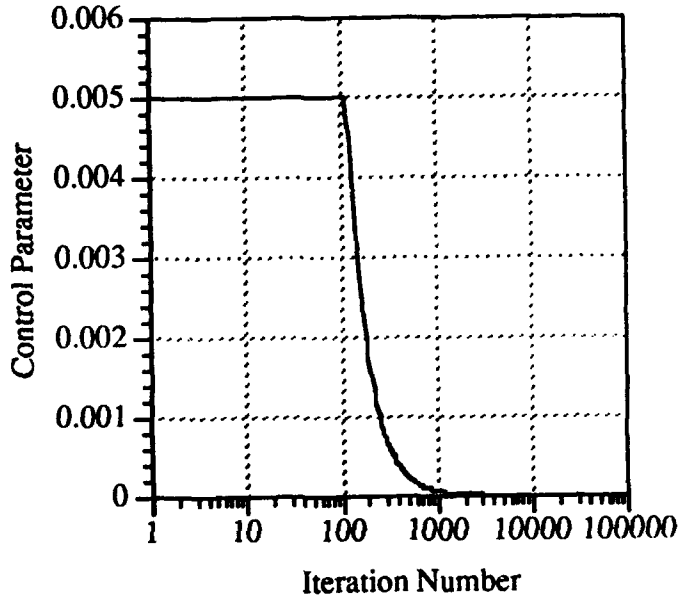


Figure 16. Control Parameter Schedule for Simulated Annealing

This control schedule was determined through a series of numerical experiments where modifications were made to the two constants A and B of Eq. 5.5 until adequate convergence of the SA algorithm was achieved for the 10 bar truss example. Initially it was found that using the actual iteration number of the simulated annealing algorithm provided good convergence. Further investigation showed a marked improvement in convergence was achieved if the acceptance number was substituted in its place. The acceptance number represents the number of updates (or modifications) that have been made to the discrete design variable vector. An update occurs when either of the acceptance criteria of Equation 5.3 were satisfied.

Figure 17 shows the relationship between the objective function change and the control factor value in the Metropolis criterion. This plot was generated from a single ACOSS II SA optimization. There were 549 occurrences during the 10,000 SA iterations where an increase in the objective function was accepted (602 cases where a decrease in the objective function was accepted). The allowable increases in the objective function decrease dramatically as the control factor is reduced. The control factor is a function of the acceptance number ; the lowest value of "c" for this case was thus $200.0(549+602)^{-2.2} = 3.69(10^{-5})$.

A simple example to illustrate the results obtained from the simulated annealing procedure was developed using the five bar truss of Figure 13. Two materials were considered for this example (refer to Table 1). A neural network was trained with the weights of the optimal designs for all 32 possible material combinations. The design space as represented by the neural network was searched by the simulated annealing procedure. The two parameters under the users control in the SA procedure (the control factor and generation mechanism) were varied so that the effect on the SA procedure could be noted. For each case investigated, the SA procedure was executed ten times, each time for a maximum of just ten iterations. The weight values shown in the Figures 18-21 are the actual neural network output values (which are in the range of 0.1 to 0.9). At the start of the SA procedure, initial weights (as shown in the figures) were obtained from a design where materials were randomly assigned to the five rods. The final weight values represent the weight of the last design accepted by the simulated annealing algorithm,.

Figure 18 shows the resulting designs obtained when the control factor was constant and equal to 0.02 and the generation mechanism involved the random selection of one of the five rods for a material change. The SA procedure was able to identify the least weight design for six of the ten designs. In every case, the procedure resulted in either a weight reduction or the initial design. In Figure 19, the control factor constant was increased to 0.2. This effectively allowed the acceptance of a larger number of increases in the objective function. This was reflected by the final designs, none of which were the least weight design. In fact, for designs 5, 6, and 8 the final objective function was actually larger than the initial weight.

Figure 20 shows the results obtained for a constant control factor of 0.02, but with a generation mechanism that changed the material for three randomly selected rods at each iteration. Six out of the ten designs were the least weight material system. Also, all the final designs are either less than or equal in weight to the initial designs. Figure 21 shows the results obtained from a control factor of 0.2 and the generation mechanism that modifies the materials of three rods at each iteration. Only one of the final designs is indeed a least weight design. In two cases, heavier designs are obtained.

The results for this simple example indicate that the value of 0.2 for the control factor was too large, with little likelihood of obtaining the least weight design evident for this particular example. If the value of C was too small, it would force the design to a local minimum, from which it cannot "escape" without an accepted increase in the objective function (unfortunately, this simple example did not exhibit this behavior). The value of 0.02 appeared more promising for this example, since the minimum design was

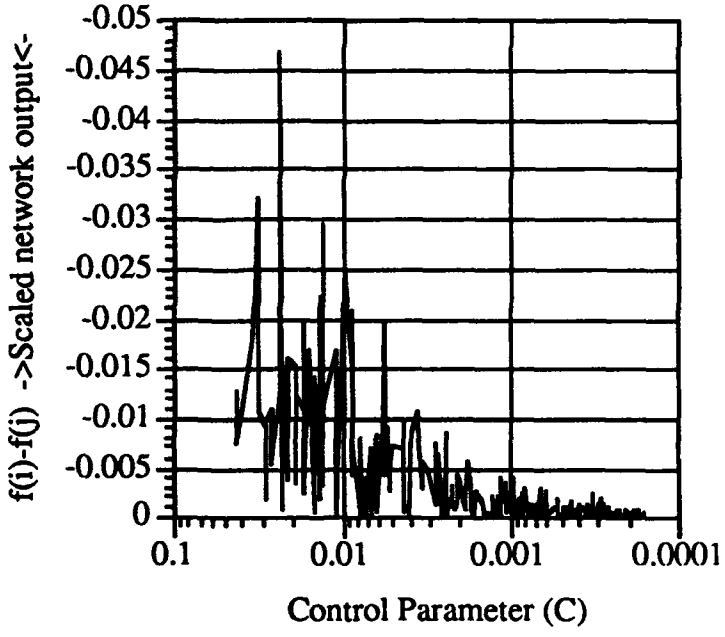


Figure 17. Change in Objective Function versus Control Parameter for the Metropolis Criterion (ACOSS II, 10,000 SA iterations)

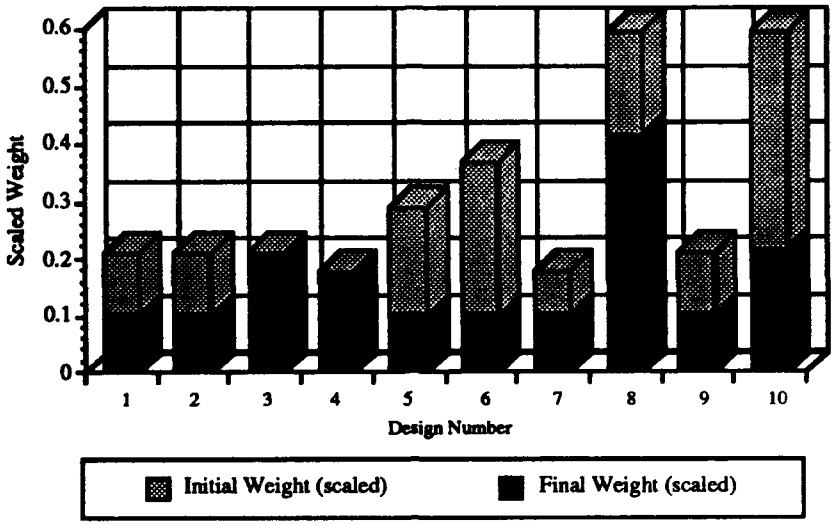


Figure 18. Initial and Final Designs Obtained for the 5 Bar Truss Material Design (C=0.02, number of rods modified=1)

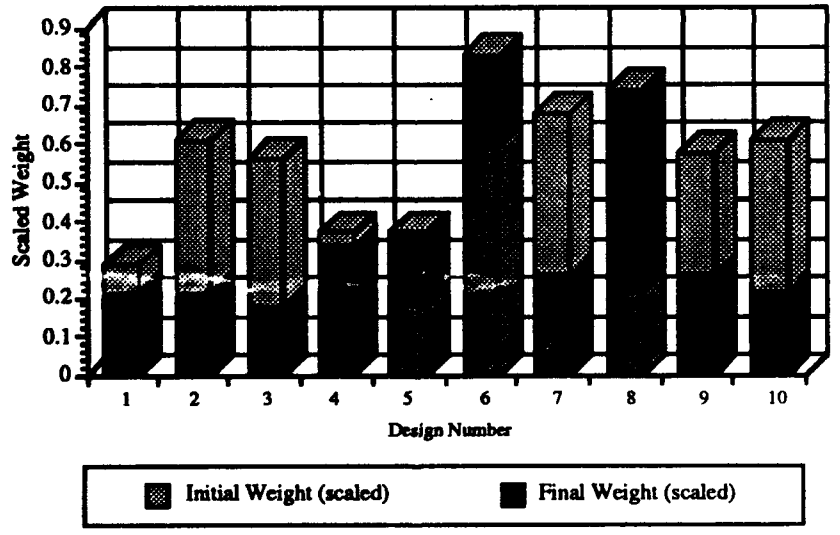


Figure 19. Initial and Final Designs Obtained for the 5 Bar Truss Material Design (C=0.2, number of rods modified=1)

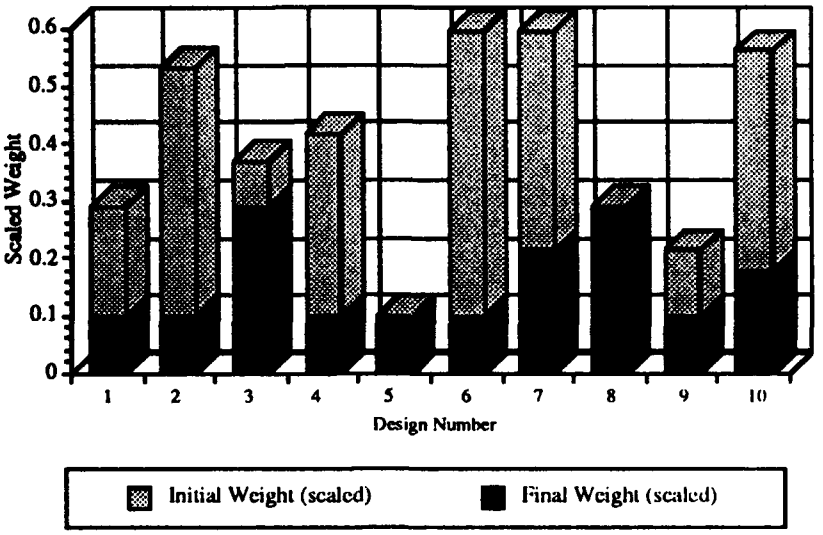


Figure 20. Initial and Final Designs Obtained for the 5 Bar Truss Material Design (C=0.02, number of rods modified=3)

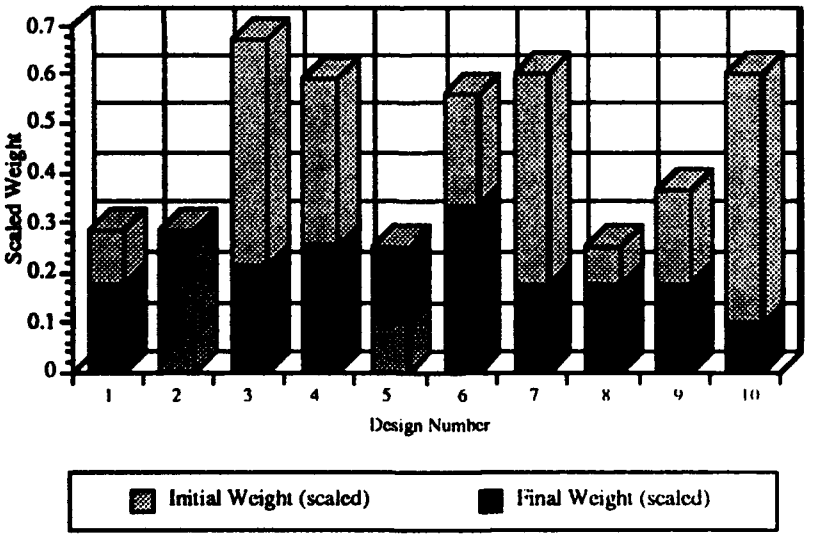


Figure 21. Initial and Final Designs Obtained for the 5 Bar Truss Material Design (C=0.2, number of rods modified=3)

selected the majority of the time. The number of rods considered by the generation mechanism seemed to have no noticeable effect on the results. The control factor and generation mechanism employed for this example could not be any simpler--constant values for each.

With the experience and insight gained with this simple example the more detailed discrete design variable problems presented in the following sections were considered.

5.4 Design for Survivability

The applications of multidisciplinary optimization and neural networks to the preliminary structural design problem as presented earlier in this section were rather generic in nature. No special design criteria was established and the overall purpose was to develop a method for quantifying the design space for a structural concept. Another potential application was considered as part of this study which had a very special purpose. The intention was to determine if one could integrate the concept of design for damage tolerance or survivability into the framework which had been established. It should be emphasized that this was a preliminary investigation of this concept and that additional efforts would be required in this area to fully exploit the potential of this approach.

The need to incorporate damage tolerance criteria into the design of aerospace structures has been recognized for many years. With the advent of multidisciplinary design optimization (MDO) and finite element analysis (FEA), structural design which takes into account possible damage conditions as simply another design constraint is a possibility. The goal of such a design would be to achieve a least-weight material distribution for the structural components so that the constraints on stress, displacement, aeroelastic phenomena, and characteristic natural frequencies, among others, are satisfied for the undamaged structure as well as all relevant damaged conditions.

Current MDO packages, while providing a powerful design tool, are typically limited to the design of a fixed geometry finite element model, where the applied loadings, and materials are also fixed. The design resulting from such an optimization procedure represents only a point in the feasible design space for the candidate structural concept. Work has been done in the area of survivable and fail-safe design which employs multidisciplinary optimization[58-60]. These approaches are applied to fixed geometry/material finite element models, so that an understanding of the design space to model variations (topology, materials, configuration geometry, etc.) was not necessarily gained.

In the current research, neural networks were applied to the problem of damage tolerant design through two distinctly different approaches. The first approach provided a mapping of a given structure's design space. The design space was composed of optimal designs based on various structural configurations and anticipated damage conditions. Optimal survivable designs were defined as least weight structures designed to meet stress, displacement, and other constraints deemed appropriate for the undamaged as well as all damaged structural conditions considered. The design space representation could then be used to define candidate structural configurations which yield the most desirable design characteristics.

In the second approach, a mapping of the damaged structural characteristics was obtained for a fixed structural configuration. The neural network was then used to predict structural behavior for any damage condition so that information about critical damage conditions could be developed. These damage conditions could then be accounted for in the design of the structure.

The application of neural networks to the design of damage tolerant or survivable structures employed in this study involves the mapping of a selected portion of the design

space of a given family of structural configurations (or their anticipated damage conditions). Once the design information has been "stored" in the neural network it then becomes a simple matter to extract the trends in optimal designs for a given set of structures. This can allow for the determination of the most desirable configuration to meet a set of design objectives. Note that the final configuration selected is often a significant improvement over the "best" configuration available in the training data for the network. This indicates an important extrapolation capability which is provided by the neural network. The prediction of structural behavior to previously unanalyzed damage conditions is also possible. These concepts are then readily applicable to a diverse set of design problems.

Figure 22 presents a flowchart for the damage tolerant design procedure used for the 3-D truss structures considered in this report. An initial analysis was performed on the undamaged structure, and then on all damage conditions under consideration. The most violated constraint (i.e. element stress level) from all the damage conditions (and the undamaged condition) was retained and used to drive the FSD design. This procedure was performed in an iterative fashion until all applicable stress constraints were satisfied (within $\pm 1\%$ of the stress allowable) or a minimum gauge constraint was reached. The applicable stress constraints were tensile allowable, compressive allowable and a local buckling constraint. This design procedure was repeated for all configurational geometries which were to be included in the neural network training data set. The procedure required approximately 20 iterations to obtain a converged solution, where each iteration involves an analysis of the undamaged as well as all damaged trusses. The procedure was robust in that convergence was always achieved, and the final designs were not affected by the initial rod areas.

For the wing-box example, an optimal baseline design was found using ASTROS. A damage criteria was developed, and the optimal baseline was then damaged (through the removal of specified finite elements) and reanalyzed to provide information on the characteristics of the damaged structure. The characteristics considered in this study were flutter speed and natural frequencies. The results of these analyses were used to provide training data for the neural network. Once trained, the neural network was used as a very efficient prediction tool for all possible damage conditions which were described by the previously used damage criteria.

Other significant aspects of this neural network approach to design are its modularity and relative computational simplicity. A neural network calculation was orders of magnitude faster than either of the finite element design procedures employed in the examples. Though a recursive approach was not taken for the damage tolerance design studies it is feasible that this could be used to help reduce the "cost" in developing the design space definition for the damage tolerant designs.

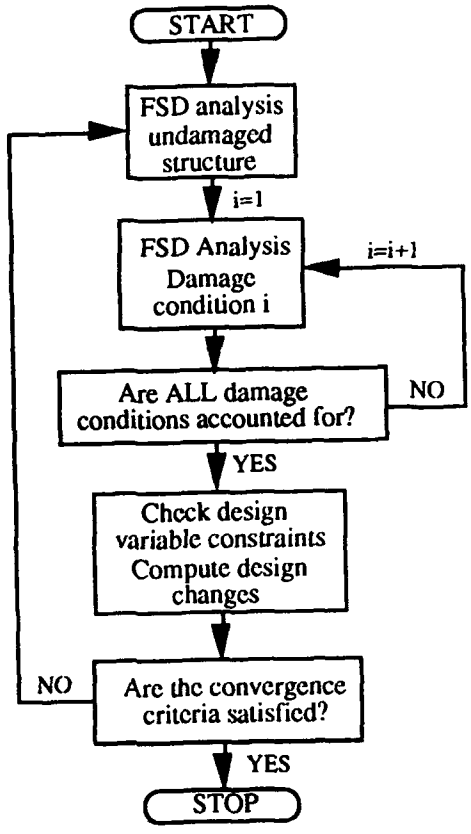


Figure 22 FSD Damage tolerant design procedure

6. PRELIMINARY DESIGN STUDIES - MODELS AND METHODS

A number of different approaches to the use of neural networks in preliminary structural design have been examined. In each case the emphasis has been to use the neural network to provide a quantitative description of the design space in a region of interest. These cases are distinguished by the nature of the design variables under consideration. The first of these groups of design studies involves continuous design variables. This variable type is illustrated through a series of configurational designs. For each of these cases the geometry of the finite element models were modified during the design. For all of the design studies involving continuous design variables, the automated fully-stressed design procedure from SWIFTOS was used.

The second group of design studies involve discrete design variables, illustrated through material system optimization. For this group of design studies, ASTROS was used. In the third group of design studies, a particular design condition was examined in some detail. A preliminary assessment of the use of neural networks to assist in damage tolerant design was conducted. For these applications both ASTROS and SWIFTOS were used as the basic source of design information. The following sections will provide a detailed description of the finite element models, design constraints, and the neural networks that have been considered for the design problems.

6.1 Configurational Design - Continuous Design Variables

Configurational design involved the determination of structural geometries that improved system performance or reduced weight. These variations in geometry were modeled by allowing certain nodes within the finite element mesh to have a range of acceptable values. The fully-stressed design procedure from SWIFTOS provided least-weight designs for each geometry considered. These designs were then employed as training data for the neural network. The neural network effectively performed a design space mapping for the configurational design variables under consideration. Once this mapping was obtained, the neural network was used as a powerful and flexible design tool.

Two examples were used to study the configurational design problem. The first was the design of a ten bar truss, where three nodes were allowed to have a range of possible values. The neural network design space representation was then employed in an optimization study where minimum weight was considered as the objective function. The second design problem involved a three spar wing-box, where the spar locations were considered as the design variables. In this study, the neural network representation of the design space was again used in an optimization study. Weight, natural frequency, and tip displacement were treated independently as the objective functions for this design study.

6.1.1 Ten Bar Truss

This study involved the configurational design of a planar truss. The truss, shown in Figure 23, was composed of ten axial force rods. The nodal coordinates of nodes 2, 4, and 5 of the finite element model were considered to be the configurational design variables. These nodal coordinates were constrained by the regions shown in Figure 23. A local Euler buckling constraint was also imposed (a solid circular cross section was assumed). It was desired to determine how the variations in these nodal coordinates would affect the optimal weight for the truss for a fixed set of applied loads.

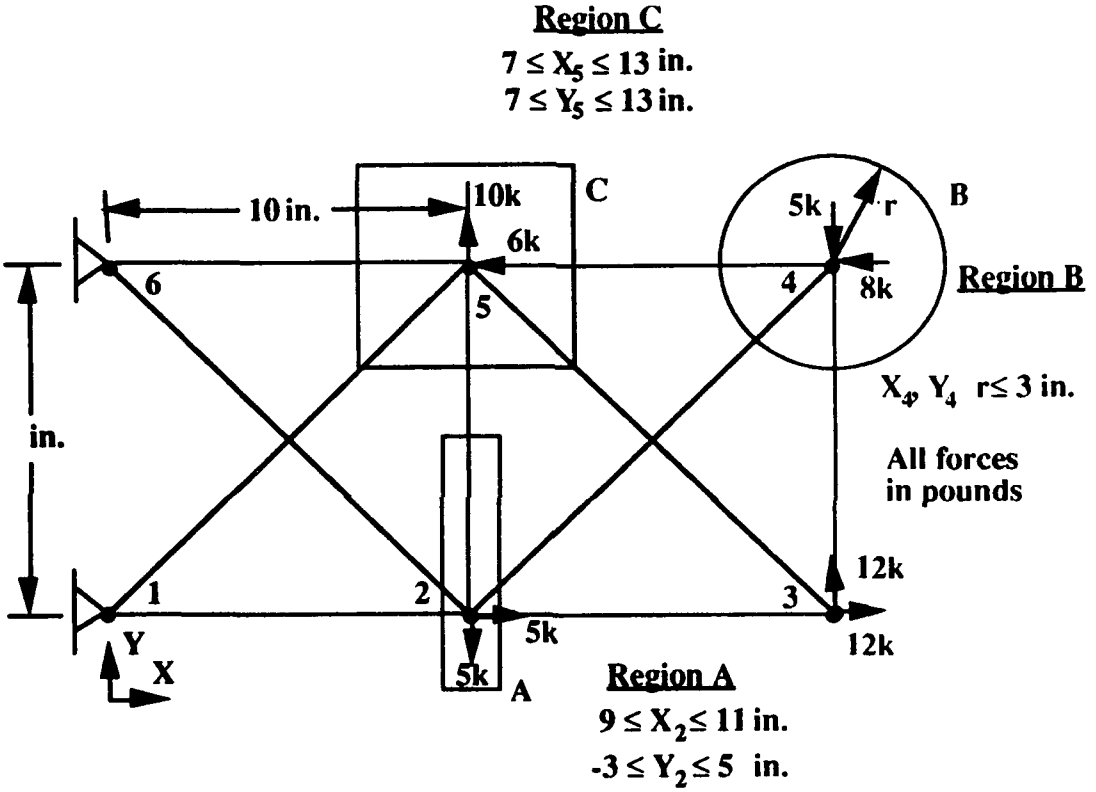


Figure 23. Ten Bar Truss, Baseline Configuration

The trained neural network was employed as a subroutine to a math-programming optimization routine so that the least weight truss geometry could be extracted from the neural network design space representation. Table 2 provides a description of the finite element model and the design conditions imposed to generate the network training data.

In order to make a preliminary evaluation of the influence of the hidden layer geometry on the design space representations, three different network hidden layer geometries were considered. These networks had 20, 40, and 80 neurons in the single hidden layer. Also, training data sets composed of 50, 100, 200, 300, 400, and 500 designs were used to evaluate network extrapolation ability as a function of the amount of available design information. Finally, two different levels of training error were considered, E= 0.02 and 0.01, to note the effect that training error had on extrapolation ability. The training error measure, E, is the difference between the actual and desired output (using the scaled-output parameter), which was then used as the termination criteria for network training. Network extrapolation ability was determined by comparing a set of 500 unique, randomly selected designs (not present in the original training data) with the network predictions for these designs. Maximum and RMS error values were used for this comparison.

This problem was formulated to provide a simple illustration of the major aspects of the neural network approach for design space representation and for the extraction of improved designs from the neural network. Unlike the five bar truss example presented earlier, this is a "six-dimensional" design space and both visual interpretation and presentation of the design space are not possible. The concepts described for this problem will lay the groundwork for discussions of the more involved design problems.

6.1.2 Three Spar Wing-box

The second configurational design problem was intended to represent a more complicated and practical design, that of a rectangular planform wing. The wing used a NACA 4412 airfoil, three spars (each with spar caps on the upper and lower surfaces), and ten ribs. All chordwise spar locations were considered as design variables. It was desired to note the influence that the spar locations had on a set of primary characteristics for the wing structure. The leading edge spar was allowed to vary between 15-30%C(local chord), the main spar was allowed to vary between 35-55%C, and the trailing edge spar was allowed to vary between 60-75%C. Spar location ranges are shown in Figure 24. It was not anticipated that the wing characteristics would be strong functions of the spar locations due to airfoil cross-section geometry but this case illustrates the manner in which other considerations can be included in the preliminary design process. If other conditions such as manufacturing, assembly, survivability, access, etc. were to be considered at this point in the process, the influence of internal geometry could be important and that influence could be described using the neural network representation of the design space.

The wing had a constant chord of 60 in., and a semi-span of 216 inches. Aluminum material properties were used for all elements in the model. A single load case was considered; a pressure load with a resultant lift equivalent to a +5g pull-up condition. The finite element model of the three spar wing-box is shown in schematic form in Figure 25.

Table 3 provides a summary of the finite element model information used for this application. Three different networks with different hidden layer geometries were considered. The hidden layers were composed of 20, 40, and 80 neurons each. Three different training sets were also used, composed of 20, 50, and 100 wing

Table 2. Finite Element Model Parameters for Ten Bar Truss Configuration Design

- 20, and 15 ksi stress allowables for tension and compression
- E=10.0(10^6) psi, ν=0.3, ρ=0.1 lb/in^3
- All rods are designed.
- FSD is employed- stress constraints only
- Initial rod area 0.1 in^2, minimum gauge 0.01 in^2
- Optimality is satisfied when elements are within 5% of the von Mises criteria
- A Euler buckling constraint (circular cross-section) is employed.
- A single load case is considered .
- Nodes 2, 4, and 5 are allowed to vary in position

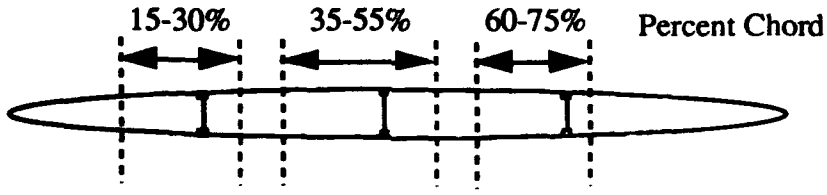


Figure 24. Cross Section Schematic of Three Spar Wing, Ranges of Possible Spar Locations

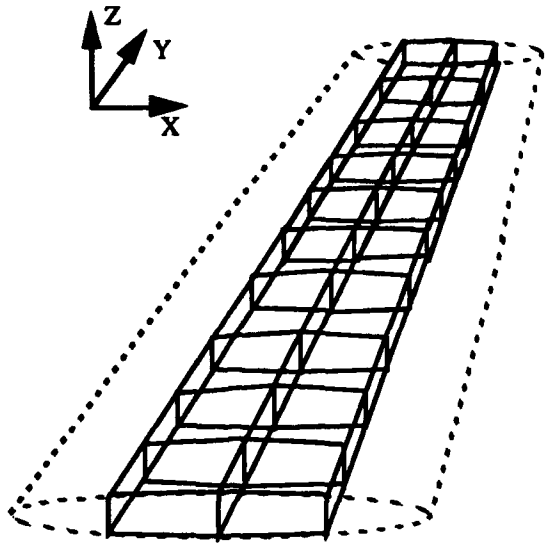


Figure 25. Schematic of Three Spar Wing Box Used for Configuration Design

Table 3. Finite Element Model Parameters for Three Spar Wing-Box Configuration Design

22.0, 19.0, and 12.0 ksi stress allowables for tension, compression, and shear respectively for 2014-T6 aluminum.
$E=10.7(10^6)$ psi, $\nu=0.3$, $\rho=0.1$ lb/in^3
All elements are designed using FSD (membrane minimum gauge of 0.01 in., rod area minimum gauge of 0.01 in^2).
Optimality is satisfied when elements are within 5% of the von Mises criteria.
No buckling constraints are imposed.
A pressure load of 7776.0 lb with a center of pressure at 36.7% C (+5g pull-up condition).
Leading edge spar ranges from 15-30% C, main spar ranges from 35-55% C, and the trailing edge spar ranges from 60-75% C.

designs. No comparisons were made between various training error levels (E=0.02 was used throughout this particular study).

The goal of this study was to perform a design space mapping of the optimal weight, tip displacement, and first three natural frequencies as a function of the spar locations. Comparisons were made of the network's ability to extrapolate from the given design information based on the network geometry and training data level. Also, the neural network design space representations were employed as subroutines to a math-programming optimization procedure so that sets of improved designs could be obtained. These designs used weight, natural frequency and tip displacement independently as the objective functions in order to further evaluate the utility of the neural network approach to extract improved designs. An improved design was considered to be one in which the objective function was lower than the "best" design present in the neural network training data.

6.2 Configurational Design Using Recursive Learning

In order to demonstrate the recursive network learning approach as applied to structural design two examples were investigated. First, a ten bar truss was considered for a configurational optimal design. A single node of the truss was allowed to range in location and the weight trends of the fully-stressed designs were noted. For this example a two-dimensional design space was developed so that a direct visual representation of the recursive process could be presented.

Second, a configurational design of a four spar subsonic light aircraft wing-box was considered. Spar locations were considered as the design variables for this example, and optimal weight, tip displacement, and natural frequency were treated independently as the objective functions and constraints. Variations in the network hidden layer geometry were not considered.

6.2.1 Ten Bar Truss

The first example was the configurational design of the ubiquitous ten bar truss shown schematically in Figure 26 along with the applied nodal loads. Table 4 provides a description of the finite element model information used for this application. This case provided a simple illustration of the recursive learning procedure. A single nodal coordinate was considered as the design variable of interest. In order to generate training data for the neural networks, both the SWIFTOS FSD procedure and the automated finite element mesh generator were used. This allowed for the automated modeling and analysis of each truss. SWIFTOS used constraints on tensile and compressive stress as well as buckling stress for the axial force rods. Figure 26 also shows the allowable limits for the node 2 location. The goal was to determine the design space defined by the least weight truss designs as a function of the location of node 2.

This example used a 2-80-80-1 network (2 inputs, 80 neurons in the first hidden layer, 80 neurons in the second hidden layer, and 1 output neuron). Preliminary studies indicated that this more involved network configuration with a second hidden layer helped to achieve an adequate convergence rate during network training (adequate convergence being defined as a continuous reduction in error down to the termination error level, within a reasonable amount of time). Acceptable convergence characteristics were important for the recursive procedure since, at each retraining cycle, the back-propagation training procedure was applied a fixed number of iterations (for this study, 10,000 iterations). If the training error satisfied the specified limit before the 10,000th iteration, training was concluded. If this did not occur, training was terminated after 10,000 iterations regardless of the error value.

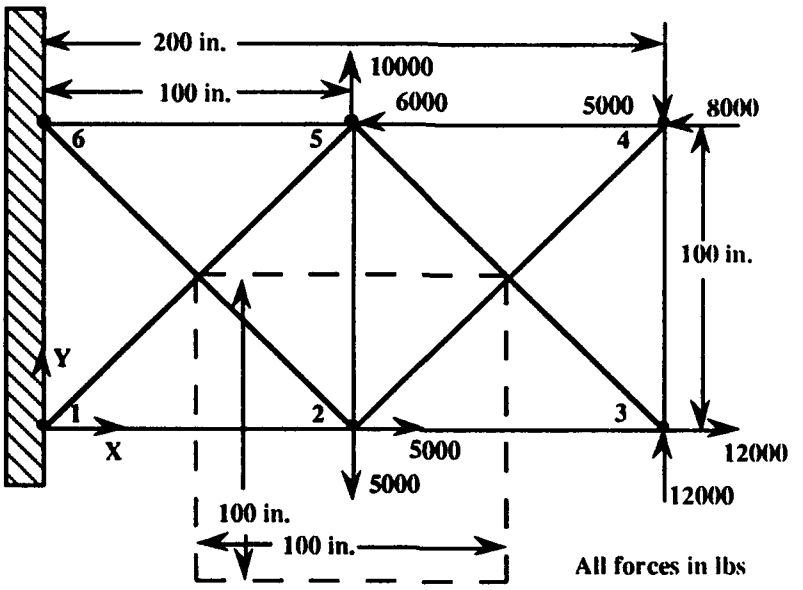


Figure 26 Schematic of Ten Bar Truss used in the Recursive Training Study, Baseline Design

Table 4. Finite Element Model Parameters for Ten Bar Truss Recursive Training Design Study

- 20, and 15 ksi stress allowables for tension and compression, respectively for mild aluminum
- $E=10.0(10^6)$ psi, $\nu=0.3$, $\rho=0.1$ lb/in^3
- All rods are designed.
- FSD is employed- stress constraints only
- Starting rod area 0.1 in^2, minimum gauge 0.01 in^2
- Optimality is satisfied when elements are within 5% of the von Mises criteria
- A Euler buckling constraint (circular cross-section) is employed.
- A single load case is considered
- Node 2 was allowed to vary in postion

This allowed for a "hands-off" approach for this study but a more active monitoring of the recursive procedure would be preferred for a practical application.

An initial training set of ten randomly selected designs was used to start the procedure. At each iteration in the recursive procedure two new designs were generated; one from the neural network predicted least weight design and the second selected at random within the design space. The incorporation of a random design helped to further refine the overall design space and also helped to reduce the likelihood of the recursive procedure isolating and refining a local minimum in lieu of a global minimum. The selection of a mechanism for the addition of candidate designs to the training set is an issue of continued interest and is probably problem dependent.

Graphical presentation of the neural network representation of the design space was accomplished by developing contour plots from data developed by propagating a 100 by 100 set of evenly spaced, allowable nodal coordinates through the neural networks. This particular problem was selected to allow for the graphical presentation of a two-dimensional design space.

6.2.2 Four Spar Wing-box

The second example involves the configurational design of a four spar wing-box where the spar locations were considered the design variables and design weight, tip displacement, and natural frequency were independently considered as the objective functions. In order to generate training data for the neural networks, SWIFTOS and the automated finite element mesh generator were used. For the wing-boxes SWIFTOS employed a von Mises constraint for each of the planar elements. Subparametric quadrilateral membranes were used to model the skin and the spar and rib webs. No buckling constraints were imposed on the wing-boxes.

The wing-box representation, Figure 27, for the constant chord wing was comparable to a model which might be used at the early stages of preliminary design. It was composed of 80 axial force rods representing the spar caps and 130 subparametric quadrilateral membranes for the skin, spar webs, and rib membranes. Table 5 provides additional details on the finite element model and associated parameters. For this example, it was desired to determine the behavior of the wing-box due to variations in the chordwise position of the four spars. The leading edge spar was allowed to vary between 15-30% C, the second spar was allowed to vary between 35-45%C, the third spar was allowed to vary between 50-60%C, and the trailing edge spar was allowed to vary between 65-75%C. A perspective on the complexity of this problem can be achieved if one considers that even if allowable designs are required to have only integer values of the spar locations (i.e. 18%, 41%, 52%, 74% would represent a single design), this design space would have over 21,000 candidate designs.

A single static loading case was used to define the basic design requirement. The loading was specified as a non-uniform pressure distribution over the planform of the wing. This produced a different set of nodal loads for every finite element mesh geometry considered.

The neural network used for this study had 4 inputs (the 4 spar locations), a single hidden layer composed of 40 nodes, and an output layer composed of 5 neurons. The five outputs were the design weight (as determined from the FSD procedure for the single static load), the corresponding tip deflection under static load, and the first, second, and third natural frequencies.

Since the goal of the study was not to develop a single point design but to provide a means for representing the concept's design space a recursive procedure was required which would help provide detail in the more critical regions of the design space. The design space cannot be "visualized" in as straightforward a

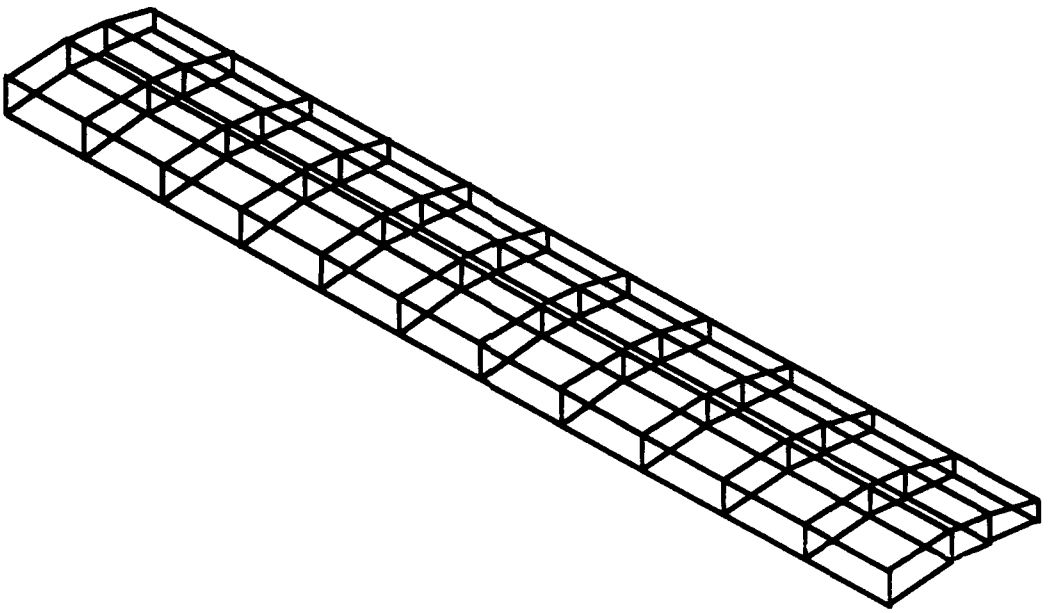


Figure 27. Schematic of Four Spar Wing-Box Used for Recursive Training Study

Table 5. Finite Element Model Parameters for Four Spar Wing-Box Recursive Training Design Study

25.0, 15.0, and 12.0 ksi stress allowables for tension, compression, and shear respectively for 2014-T6 aluminum.
E=10.5(10^6) psi, ν=0.3, ρ=0.1 lb/in^3
All elements are designed using FSD (membrane minimum gauge of 0.01 in., rod area minimum gauge of 0.01 in^2).
Optimality is satisfied when elements are within 5% of the von Mises criteria.
No buckling constraints are imposed.
A pressure load of 7776.0 lb with a center of pressure at 36.7% C(+5g pull-up condition).
Spar 1 ranges from 15-30% C, spar 2 ranges from 35-45% C, spar 3 ranges from 50-60% C, and spar 4 ranges from 65-75%C.

manner as the previous example. The network was trained with a baseline set of data composed of only two designs. These two designs were selected at random from a subset of 32 designs uniformly distributed over the allowable design space.

The five network outputs were used in a cyclic fashion as independent objective functions to drive the recursive training. Tip displacement was considered as the first objective function, followed by weight and then the three natural frequencies. The subsequent optimization provided a new candidate configuration which was designed using the FSD procedure and added to the training set. After the third natural frequency was used, the weight again became the objective function. Twenty-five iterations were performed, so that each network output was used as the objective function five times. Each of the 25 optimal design predictions obtained from the neural network was used as additional training data to further refine the design space. A comparison between the recursively predicted optimum values and two other network predictions was also performed. In the first case 27 randomly selected candidate designs were used, this is the same size as the training set developed using the recursive procedure. In the second case 100 randomly selected designs were used in the training set.

6.3 Material Selection - Discrete Design Variables

Discrete design variables must be considered in a fashion different from continuous design variables. In particular, gradient-based search techniques cannot be used to identify optimal designs involving these variables. Unconventional approaches, such as genetic algorithms or the simulated annealing algorithm have therefore been developed. To investigate the utility of the neural network approach for discrete design variables two different design studies were considered. The first problem was the simple planar ten bar truss in which four isotropic materials were considered for each of the ten bars. The second example was an ACOSS II space truss, where four materials were considered for each of the 113 rod elements. ASTROS was used to generate the network training data. Recursive learning was not attempted for the discrete variable design problems.

To implement the simulated annealing algorithm a control parameter schedule and generation mechanism were determined. The control parameter schedule determines how the control parameter (used for the acceptance criteria) varies as the simulated annealing algorithm iterations progressed. The complexity of the schedule reflects the complexity of the combinatorial optimization problem. In the examples presented here, the simple power function relationship shown in Figure 28 was used as the control parameter schedule.

The development of this control parameter schedule was described in Section 5.3.2.3. Another implementation issue deals with determining an appropriate generation mechanism for the design problem. The generation mechanism describes the manner in which the discrete design vector was perturbed between successive iterations. As with the control parameter schedule, the generation mechanism's complexity will reflect the complexity of the combinatorial optimization problem. Initially, a mechanism that involved the random selection of a number of elements along with a random material change for the discrete values of these elements was performed. The number of elements selected for material change was initially variable; starting as a large percentage of the total design variables, and slowly decreasing as the number of iterations increased. This was further modified when the new control parameter schedule was adopted. The new generation mechanism involved a fixed number of elements being perturbed at random. In the case of the ten bar truss, two elements were selected at random, and then their material values were assigned a random value. For the ACOSS II design, five rods were selected at random, each with a randomly assigned material value. Improved convergence was

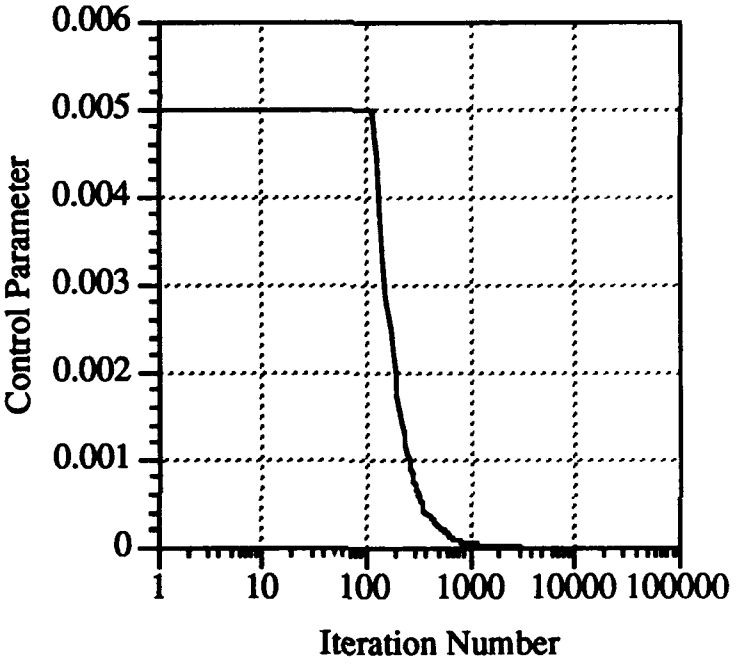


Figure 28. Simulated Annealing Control Parameter Schedule

noted over the original generation mechanism, even though the new mechanisms were considerably simpler.

Because of the efficiency of the neural network, 10^{-3} CPU seconds per iteration were required (based on a DECstation 3100) for the neural network approximate solution to be generated, it was not critical that the control parameter schedule and generation mechanism be "efficient"; many thousands of SA iterations could be performed in a short period, countering the need for efficiency in the generation mechanisms. As the neural network complexity increased (i.e., more hidden layers or more neurons per hidden layer), computational times would rise. An increase in network complexity may require the development of more efficient schedules.

6.3.1 Ten Bar Truss

The first example discrete design variable problem was the ten bar truss, shown in Figure 29. This simple model was employed to determine if the neural networks could effectively represent the discrete design space, and if so, to find the distribution of material types for the truss that would minimize the truss design weight. Four isotropic materials were considered for each of the ten rods resulting in 4^{10} possible material configurations.

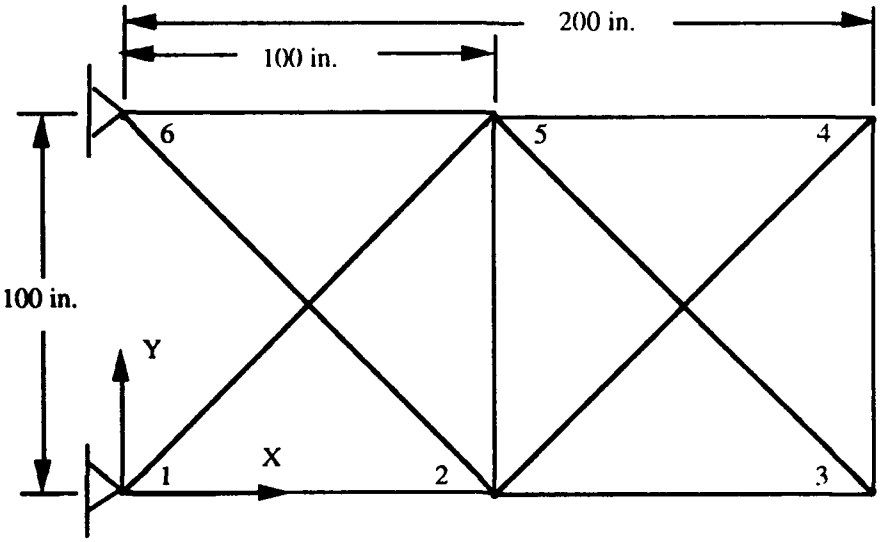
Table 6 provides a description of the finite element model information used for this example. The properties for the materials considered for the material system are listed in Table 7. Two load conditions were considered and an upper bound natural frequency constraint on the first mode of 30 Hz was applied. Designs were obtained using ASTROS. No buckling constraints were imposed.

Four different neural networks were considered to determine what effect, if any, neural network geometry would have on the results for this design space composed of discrete design variables. Each of the four had 40 inputs, and output layers with 1 neuron. The networks had a single hidden layer composed of 20, 40, 80, or 120 neurons. Four different sets of training data were also used, composed of 20, 50, 100, and 200 input/output pairs in order to make determinations regarding network accuracy, learning times, and extrapolation ability based on training data detail.

The material vectors were developed in the following manner: each rod element had the possibility of being composed of one of the four materials shown in Table 7, consequently there were four network inputs for each rod element. Only one of these four inputs had a value of 0.9 and that input corresponded to the "active" material assignment. The other three inputs had a value of 0.1. Figure 30 describes the initial generation mechanism employed for this design study. This figure is for illustrative purposes only; it is meant to describe one approach to a generation mechanism which was considered in this study. This figure indicates the number of rods that were allowed to change material properties in any given iteration. Rods and corresponding material changes were selected at random. Note that the variations made to the design vector were continually reduced as the design approached the global minimum. The generation mechanism that was actually employed for the results presented herein was much simpler since only two rods were allowed to vary at any given iteration. This extremely simple generation mechanism yielded excellent convergence results as is discussed in the following section.

6.3.2 ACOSS II

The second discrete design variable example is an ACOSS II space truss, developed by the Charles Stark Draper Laboratory[18]. The finite element model,



Node	Load Case I (X, Y -->kips)	Load Case II (X, Y -->kips)
2	5, -5	-8, 3
3	12, 12	-12, -2
4	-8, -5	5, 8
5	-6, 10	-10, -6

Figure 29. Ten Bar Truss Model and Applied Loads, Discrete Design Variable Application

Table 6. Finite Element Model Parameters for Ten Bar Truss Material System Design Study

Four different materials are considered
$E=10.0(10^6)$ psi, $\nu=0.3$, $\rho=0.1$ lb/in^3
All rods are designed.
ASTROS - math-programming is employed- stress constraints and a natural frequency constraint (1st freq.<30 Hz)
Starting rod area 0.5 in^2, minimum gauge 0.01 in^2
Optimality is satisfied when all applicable constraints are met and the change in the objective function is less than .025% for two consecutive iterations
No buckling constraints are imposed (none available)
Two different load cases are considered

Table 7. Material Properties for Ten Bar Truss Design

Material	Modulus (psi)	Density (lb/in^3)	Tensile Allowable (psi)	Compressive Allowable (psi)
1) 2024-T3	$10.7(10^6)$	.100	$20.(10^3)$	$15.(10^3)$
2) 7075-T6	$10.5(10^6)$	.101	$36.(10^3)$	$37.(10^3)$
3) Steel	$28.5(10^6)$	.283	$85.(10^3)$	$71.(10^3)$
4)Titanium	$17.0(10^6)$	.164	$75(10^3)$	$75.(10^3)$

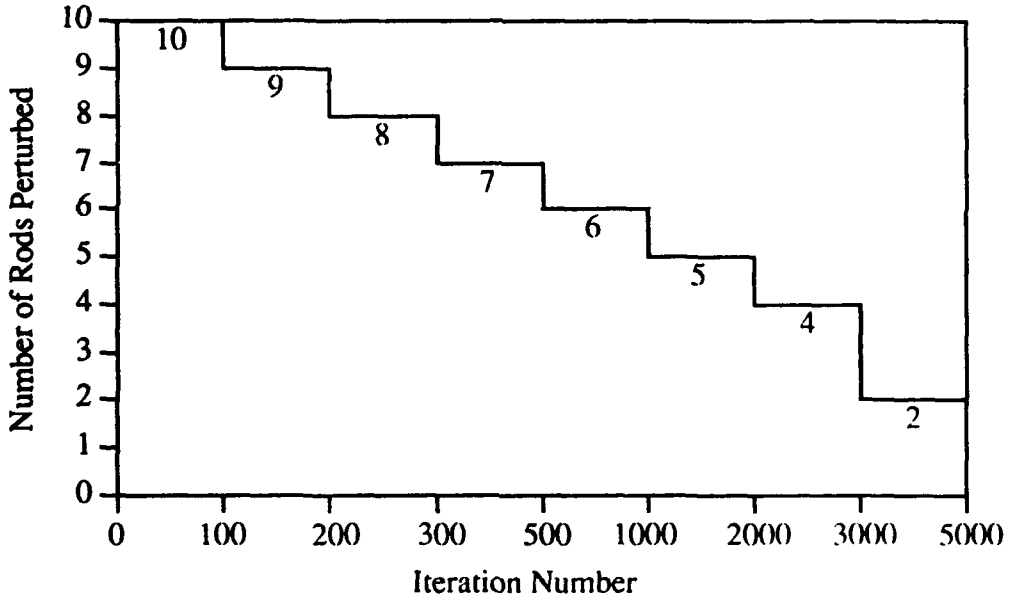


Figure 30. Generation Mechanism Schedule for the Ten Bar Truss

shown in Figure 31, was composed of 33 nodes (90 degrees of freedom), 113 axial force rods and 18 concentrated (non-structural) masses totaling 11,217.2 lb. As in the first example, it was desired to determine the material configuration that would provide the least weight design for the truss (subject to the frequency constraints). ASTROS was used to generate the design information for this study.

Design constraints were placed on natural frequency only; a lower bound constraint of 2.0 Hz on the first mode and 3.0 Hz on the second mode were used. Four materials were considered; these materials are listed in Table 8. Each of the 113 rods could be composed of any one of the four materials, resulting in 4^{113} possible material combinations. The material vector (used for neural network input) was developed in the same manner as for the ten bar truss case. That is, there were four network inputs corresponding to each of the 113 rods, and only one of the four inputs had a value of 0.9, the other three had values of 0.1.

Parameters for the ACOSSII finite element model are given in Table 9. As in the ten bar truss problem, the initial generation mechanism had a gradually decaying number of rods which were perturbed between successive iterations. A simpler generation mechanism was substituted. This mechanism had only a constant number of rods, five, being perturbed between iterations. Improved convergence characteristics were noted for this mechanism and the results presented in this report used this generation mechanism.

Three different network geometries were considered, involving 10, 40, and 100 neurons in the hidden layers. Training sets of 50, 100, 300, and 500 ASTROS designs were used. Determining the effects that hidden layer geometry and IOP size had on the neural network's representation of the design space were the primary objectives of this study.

6.3.3 Multi-Spar Composite Wing

The final discrete design variable example was an intermediate complexity wing, again taken from the ASTROS Applications Manual[3]. In comparison with the other design studies this was rather superficial and is included to indicate the potential of other types of applications which could occur during the preliminary structural design process. In this application, the laminae orientations for the composite wing skin could have been either a continuous or a discrete design variable. As indicated below this required the development of an alternative method for the design variable description. The selection of orientation as a discrete design variable might be that indicative of a design decision driven by a manufacturing consideration.

A schematic of the finite element model of the wing is shown in Figure 32 The finite element model was composed of 39 rods, 55 shear panels (used to model the sub-structure), 62 quadrilateral elements and 2 triangular elements to model the composite skins. The element used to model the skins was an isoparametric membrane-bending element whose characteristics are described in Reference 3. The substructure composed of the shear panel spar and rib webs and connecting posts were made of isotropic material and the skins were composite. Material properties for this finite element representation are presented in Table 10.

Each skin element was composed of four orthotropic laminae of equal thickness. The orientation of the four laminae within each skin element were used as the discrete design variables in the neural network design study. It was desired to determine the four lamina orientations that would provide the lowest first natural frequency for the structure which had been "optimized" to achieve least weight and satisfy a number of static and dynamic constraints. ASTROS was used to provide the

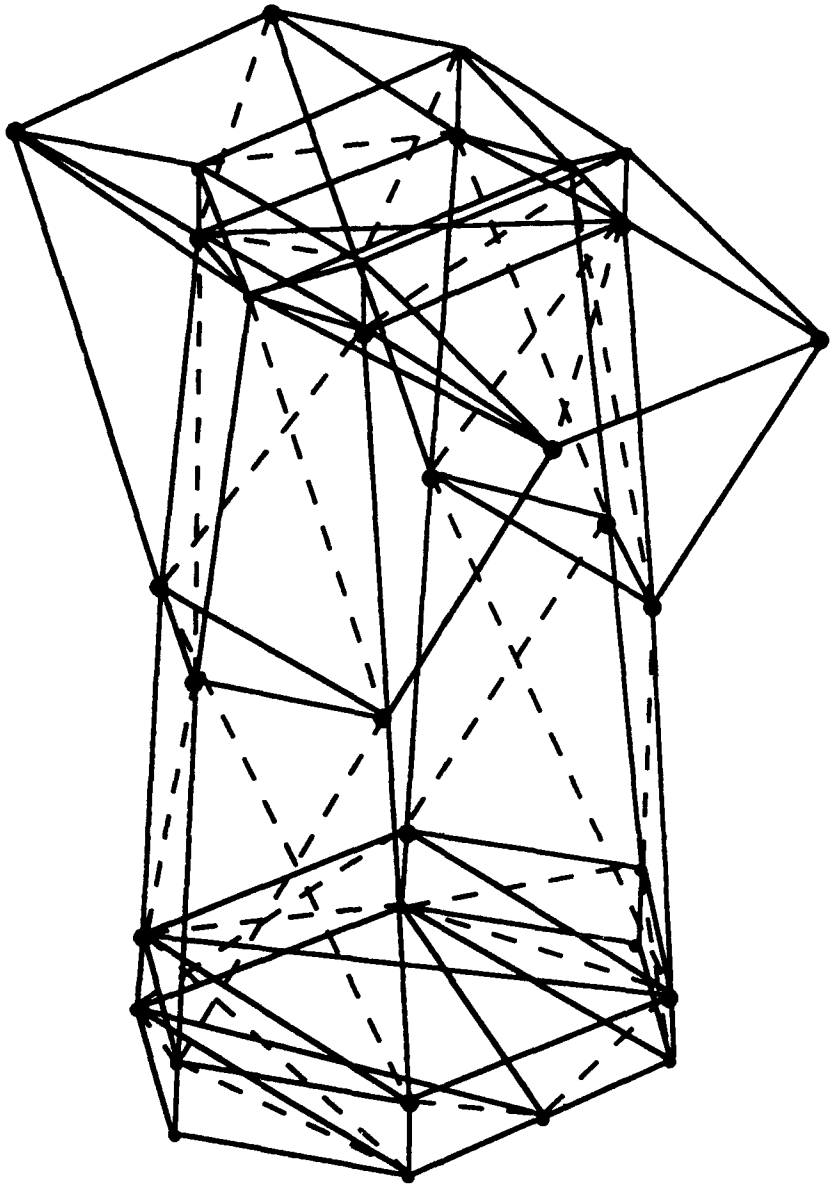


Figure 31. Schematic of ACOSS II Space Truss Finite Element Model (Adapted from Reference 3)

Table 8. Material Properties for ACOSS II Truss Design

Material	Modulus (psi)	Poisson's Ratio	Density (lb/in^3)
G ıite/epoxy	18.5(10^6)	0.28	0.0547
F ı/epoxy	32.5(10^6)	0.23	0.0723
Graphite/epoxy type I	30.9(10^6)	0.28	0.0576
Graphite/epoxy type II	21.5(10^6)	0.28	0.0526

Table 9. Finite Element Model Parameters for ACOSS II Truss Material System Design Study

Four different materials are considered
All rods are designed.
ASTROS - math-programming is employed- two natural frequency constraints only(1st freq.< 2 Hz, 2nd freq. < 3 Hz)
Starting rod area 0.5 in^2, minimum gauge 0.01 in^2
Optimality is satisfied when the frequency constraints are met and the change in the objective function is less than .025% for two consecutive iterations
No buckling constraints are imposed (none available).

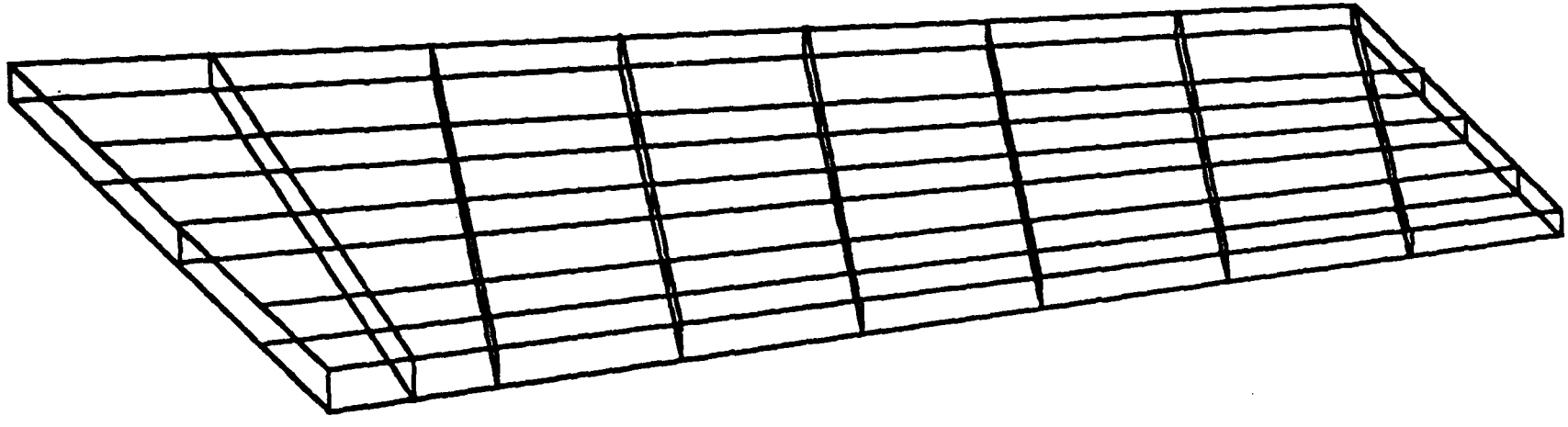


Figure 32. Schematic of Intermediate Complexity Wing Finite Element Model

Table 10. Material properties for the ICW design

Isotropic material	Orthotropic material
$E = 10.5(10^6)$ psi	$E_1 = 18.5(10^6)$ psi
$\nu = 0.30$	$E_2 = 1.6(10^6)$ psi
$\rho = 0.10$ lb./in^3	$G_{12} = 0.65(10^6)$
$t_{min} = 0.02$ in	$\nu_{12} = 0.25$
$\sigma_T \leq 67$ ksi	$\rho = 0.055$ lb/in^3
$\sigma_c \leq 57$ ksi	$t_{min} = 0.00525$
$\tau_{xy} \leq 39$ ksi	$\lvert\sigma_x\rvert \leq 115$ ksi
	$\lvert\sigma_y\rvert \leq 115$ ksi

"optimum" designs for a given set of ply orientations. For a given ASTROS design the ply orientations were the same for all of the skin elements.

Each of the four lamina orientations were allowed to range from -45° to 90°, in 5° increments. The training data was developed by randomly selecting one hundred sets of skin orientations for which the lamina orientations were chosen from the allowable design set. Each of these wings were optimized using ASTROS for minimum weight. Constraints for the optimization were placed on the displacement of the wing tip nodes (± 10 in.), flutter speed(925 KEAS @M=.8), as well as a stress constraints for two static loads. These 100 designs provided the training data for a neural network. The neural network was composed of 24 inputs, 50 nodes in the single hidden layer, and a single output neuron corresponding to the first natural frequency of the "optimized" wing.

The discrete character of the laminae orientation design variable was treated in a special fashion for this study. The angular orientations were discretized to provide a network input with a "binary-like" representation. Each lamina orientation required six "bits" of "coded" data to describe the lamina orientation. A "low" value for the bit was 0.1 and a "high" value was 0.9. The lead "bit" for each lamina was used to signify sign (.9 = "+", .1 = "-"). The following illustrates the scheme used to represent the discrete lamina orientations.

lamina orientation	"binary" code	neural network input
0°	0, 0, 0, 0, 0, 0	0.1, 0.1, 0.1, 0.1, 0.1, 0.1
5°	0, 0, 0, 0, 0, 1	0.1, 0.1, 0.1, 0.1, 0.1, 0.9
-5°	1, 0, 0, 0, 0, 1	0.9, 0.1, 0.1, 0.1, 0.1, 0.9
10°	0, 0, 0, 0, 1, 0	0.1, 0.1, 0.1, 0.1, 0.9, 0.1
-10°	1, 0, 0, 0, 1, 0	0.9, 0.1, 0.1, 0.1, 0.9, 0.1
15°	0, 0, 0, 0, 1, 1	0.1, 0.1, 0.1, 0.1, 0.9, 0.9

etc.

This representation required then required 24 total inputs to define the orientation of the four plys for the composite skin.

6.4 Design for Survivability

Two different types of neural network applications to damage tolerant design are presented. In the first example, the neural network procedure was used to determine the effect that configurational layout of a helicopter tail-boom has on the weight and natural frequency of a damage tolerant structure. A single, fixed, applied loading and six different damage conditions were used in the design, and structural configurations were obtained to achieve minimum weight. In the second example, a finite element model of a wing-box was considered for a damage tolerance study, and the trained neural network was used as a prediction tool for the structural response (natural frequency, flutter occurrence and flutter speed) of the wing to various damage conditions.

SWIFTOS was used to generate training data for the neural network in the first example. This allowed for the efficient design of hundreds of trusses of various geometries and damage conditions in an automated fashion. Constraints on tensile and compressive stress as well as buckling stress for the axial force rods were considered. Multiple damage conditions were included, with the most violated constraint of each element for the undamaged and damaged condition(s) being retained to drive the FSD algorithm.

For the damage tolerant lifting surface application ASTROS was used to analyze a set of damaged wing-boxes. This was augmented with the automated pre-processor which was used to develop the ASTROS bulk data. These wing-boxes,

whose element gauges were determined through a stress and flutter constrained optimization on an undamaged baseline wing, were subjected to various possible damage conditions. The automated mesh generator was used to select damage conditions at random.

6.4.1 Helicopter Tail-Boom

A helicopter tail-boom was considered for a damage tolerant configurational design[60]. A structure of this type can be modeled in two ways; a closed tail-boom with a stressed-skin cover over the longerons and cross members or an open tail-boom consisting only truss elements. The latter was employed in this study. Figure 33 shows the tail-boom truss. The regions indicated by sections A-D were parts of the structure where certain nodes were allowed to vary in longitudinal location. The range of locations for the nodes that make up the lateral "frame" at each cross section location are shown in Figure 33. Figure 34 shows the loads applied to the structure. These loads were used for the undamaged structure as well as all the damage conditions considered for this study.

The finite element model is composed of 108 axial force rods. A summary of the finite element model parameters is given in Table 11. A local buckling stress constraint was used, but global buckling behavior was not considered. No design variable linking was used, thus there were a total of 108 unique design variables for the undamaged, baseline configuration. This configuration is similar to a previous optimization study presented in Reference 60.

It was desired to determine the effect that configurational changes would have on optimal weight and natural frequencies for the given loading. Six possible damage conditions were considered. These were the same damage conditions used in Reference 60. The damage conditions consisted of the removal of a single node, and the nine rod elements connected to it. The configurational design variables involve the locations of sections A, B, C, and D, which were allowed to vary in the X_1 direction. Modifying these section locations effectively modifies all three coordinate values for the applicable nodes at each section. Sections vary in a manner such that the overall geometry of tail-boom remains linear from root to tip.

Section A was allowed to range in X_1 value from 66.5 in. $\pm$10.0 in.
Section B was allowed to range in X_1 value from 99.5 in. $\pm$10.0 in.
Section C was allowed to range in X_1 value from 127.5 in. $\pm$ 6.0 in.
Section D was allowed to range in X_1 value from 151.5 in. $\pm$ 5.0 in.

One hundred, least weight trusses were designed using SWIFTOS, where the location of Sections A, B, C, and D were selected at random. The resulting designs were analyzed to determine the first three natural frequencies and weight. A neural network composed of 4 inputs, 48 nodes in the single hidden layer, and 4 outputs (weight, first three natural frequencies) was trained to within 1% error using this training data.

6.4.2 Five Spar Wing-Box

A built-up finite element model of a wing-box was also considered to illustrate the utility of this neural network application for more advanced damage tolerance design applications. This application is different than the first in that rather than forming a mapping of a set of continuous design variables the neural net now was working with a set of discrete variables.

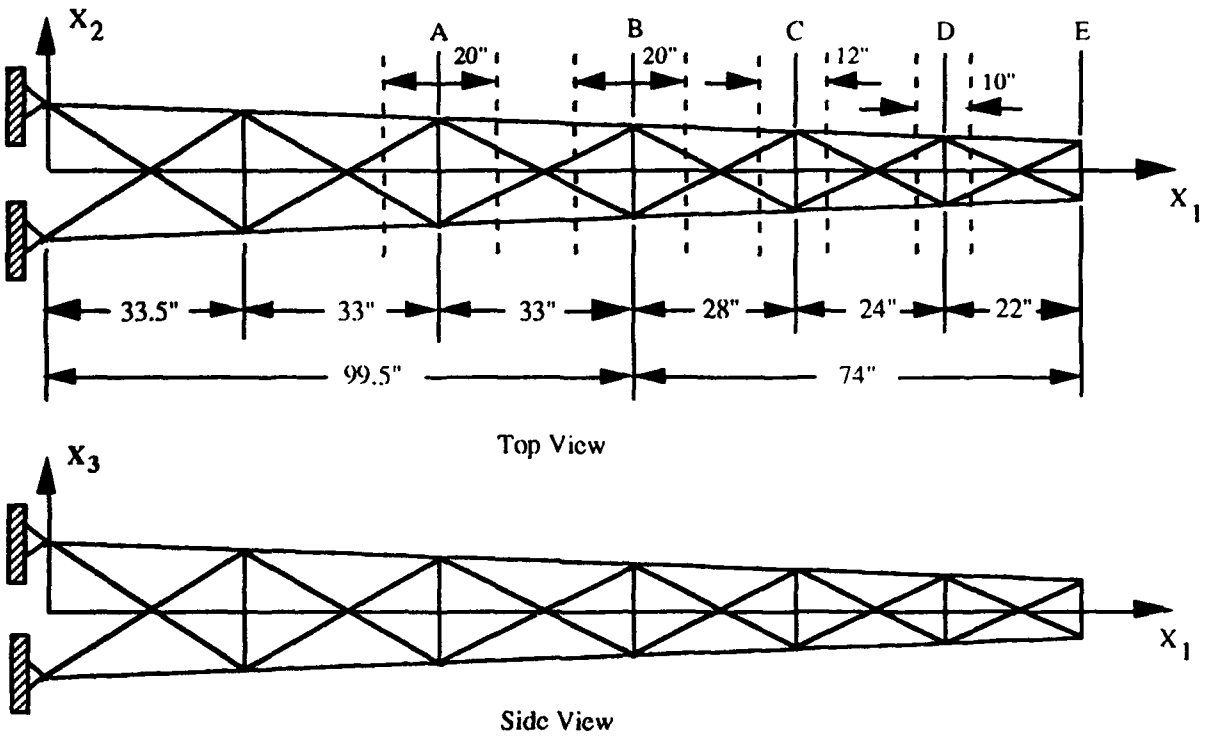


Figure 33. Helicopter Tail-Boom Used in Damage Tolerant Design Study

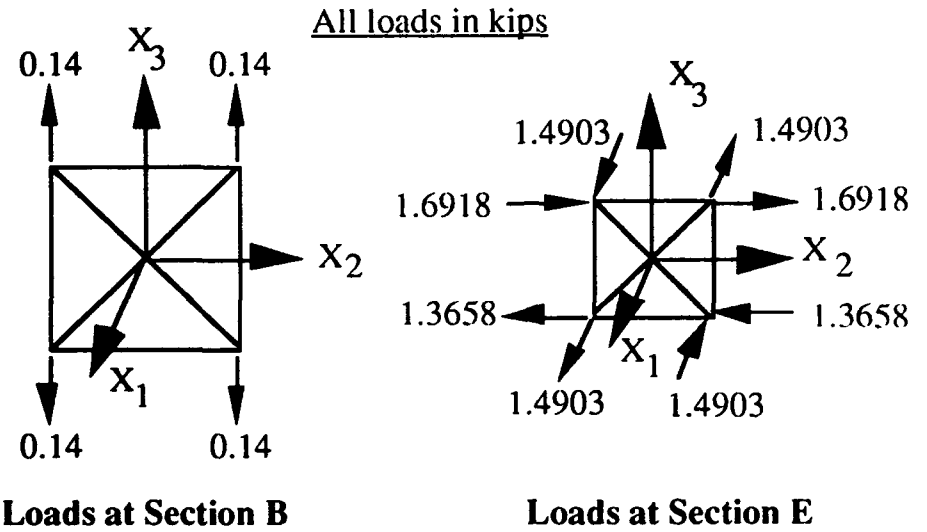


Figure 34. Applied Loads for Helicopter Damage Tolerance Study

Table 11. Finite Element Model Parameters for Helicopter Tail-Boom Damage Tolerance Design Study

Solid and circular in cross-section rod elements
Constructed of 2024-T3 aluminum alloy
E= $10.5(10^3)$ ksi, ρ= 0.1 lb/in.^{3}
Stress allowables of $\pm$25 ksi.
Minimum allowable cross-sectional area of .0415 in.^{2}.

The wing under consideration had a root chord of 100.0 in., a tip chord of 50.0 in., and a semi-span of 300.0 in. A NACA 0010-34 airfoil was used to define the nodal coordinate locations, and spars were located at 15.0, 30.0, 45.0, 65.0, and 80.0% chord. Non-structural mass totaling 800 lb was distributed over the wing. The finite element model, along with a typical damage case, is shown in Figure 35. The undamaged model was composed of 50 axial force rods representing the connecting posts, 90 shear panels representing the spar and rib membranes, and 80 isoparametric membranes representing the skins elements. The wing was cantilevered at its root. The baseline wing was designed for two loading conditions, a +4g pull-up and a -2g condition, as well as a flutter constraint of M=.85 at sea level was imposed. The design model was composed of 131 variables (all connecting posts were physically linked, all spar shear panels were unique, and all skin membranes were unique design variables). Once the baseline design was obtained, 150 ASTROS analyses were performed on damaged models.

For this example, it was desired to determine the behavior of the wing-box to possible "moderate" damage conditions. Moderate damage was defined as the complete loss of stiffness contribution for two spar shear panels and six skin membranes. The location of these elements was then selected at random over the structure for each of the 150 models. This damage model does not exclude the possibility of localized or "concentrated" damage, which is more typical of ballistic damage models. Those elements considered damaged were completely removed from the finite element model.

For the damaged wings, a modal analysis was performed to determine the dynamic characteristics and a flutter analysis was performed to determine if the damaged wings would experience flutter (recall that a flutter constraint was imposed on the design of the baseline wing). The analysis of each damaged wing required roughly one CPU hour on a DECstation 3100. The inputs to the neural network were an identifier for the damaged elements of the structure, and the neural network outputs were the structural characteristics for the corresponding damage condition. A neural network composed of 130 inputs (representing all spar shear web elements and all skin membranes) a single hidden layer of 24 neurons, and 5 output neurons (representing flutter likelihood, associated flutter speed, and the first three natural frequencies) was trained to within 1% using all of the 150 cases of the ASTROS training data. The input vector described the current state of the structure; non damaged elements had a value of 1.0 in the input vector while damaged elements had a value of 0.0. An additional 50 damaged wings were also analyzed for comparison with the neural networks prediction of their behavior.

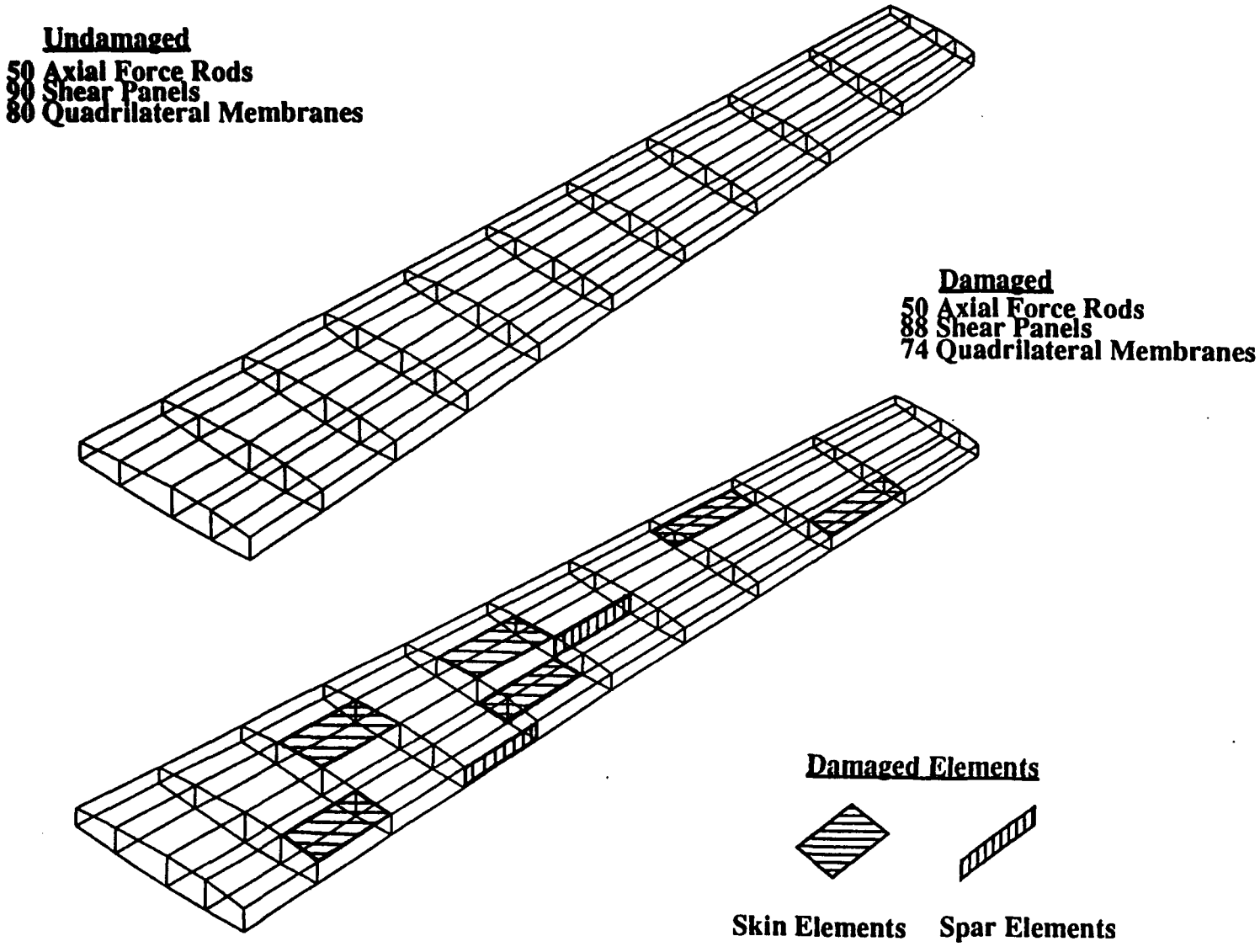


Figure 35. Five Spar Wing Box and Damage Condition, Damage Tolerance Study

7. PRELIMINARY DESIGN STUDIES - RESULTS AND DISCUSSION

This section presents the results of the design studies outlined in Section 6. The studies are organized into four sub-sections. They involve:

1. Configurational design with continuous design variables
2. Recursive training for continuous design variables
3. Material selection with discrete design variables
4. Damage tolerant design.

In each case a quantitative representation of the design space using neural networks was developed, and the influence that both neural network geometry and training had on the final designs was determined. In certain cases the character of the design space was also determined. In those cases where the neural networks were used as part of an optimization problem, improved designs were obtained.

7.1 Configurational Design - Continuous Design Variables

The discussion in this section centers on the configurational design examples, those involving continuous design variables. The nature of the design space representations as well as the extent to which improved designs were obtained is discussed. In each of the problems, the final goal was to obtain results that would indicate useful information for informed preliminary design decisions.

7.1.1. Ten Bar Truss

For the ten bar truss model, neural network training data was formulated using FSD weight values for randomly selected truss geometries. Two different neural networks and six different sets of training data were used. The two neural networks were distinguished by the number of neurons in the single hidden layer. The hidden layers had 20 and 40 neurons, respectively. The sets of training data considered involved 50, 100, 200, 300, 400, and 500 designs. Two different levels of training were used. In the first set, the training termination criterion was set to E=0.02, while in the second set the training level was E=0.01. Recall that E is the difference between the desired and the actual outputs using scaled output variables.

In order to evaluate the ability of the various network and training data combinations to effectively represent the design space, a set of 500 additional truss designs was created and used for comparison purposes. None of these designs were present in the original data used for the network training, and all 500 had randomly selected geometries. The network predicted weight values were compared to the actual weight values, and a maximum error value for the entire set of 500 designs was obtained. An RMS error value was also calculated. Maximum error was simply the highest percent error, where percent error is defined by

$$\%error = \left| \frac{(actual - network)}{actual} \right| \times 100. \tag{7.1}$$

where the actual values were data from the additional set of 500 designs and the network values were the neural network outputs.

The RMS error was defined by the following equation

$$RMS\,error = \sqrt{\frac{\sum_{i=1}^{n} \%error_i^2}{n}} \tag{7.2}$$

where n is the number of propagated data sets (n=500).

Figure 36 shows a comparison between the maximum error values for the two networks, where maximum error is simply the highest percentage error obtained from any given network prediction for the 500 new designs. The neural networks used in this figure had a training error value of E=0.02. The numbers shown above the columns in the figure represent the design designation (an arbitrary designation for each of the 500 new designs) which produced the maximum error value shown.

There is an interesting distinction that can be made about the 6-40-1 network in contrast to the 6-20-1 network. For the smaller training set size cases of 50 and 100 IOPs (Input-Output Pairs), there is a marked difference in maximum percent error. In the case of the 50 IOP networks, there is nearly a 2 to 1 difference in error between the 6-20-1 and 6-40-1 networks, respectively.

Table 12 shows designs that produced the maximum errors. The design number in Table 12 corresponds to the numbers above the columns shown in Figure 36. These designs seem well dispersed in the design space, although the Y_5 value does have only a small range of values (7.242-8.052 in.). This would imply that training data located in this region was sparse. Note also that the design numbers shown in Figure 36 do not remain constant but vary as the IOP number increases. This is because certain regions of the design space were becoming better refined with the additional IOP sets. This shifted the maximum error to other points in the design space.

Another distinction that can be made about the two networks is that the 6-40-1 network exhibited a much more consistent maximum error value as the IOP number increased (in the range of 15%). The 6-20-1 network, in contrast, showed a definite trend toward a reduction of maximum error as the IOP number increased. The 6-20-1 network with 500 IOP's also exhibited the lowest maximum error value, 11%.

Finally, the 6-40-1 network exhibited a lower average percent error; 15.3% as compared to the 6-20-1 network's value of 18.7%. The maximum error trends refer to individual points in the design space (outside of the training data), and this may not be the best indicator of the overall network performance. The RMS error may be a more valuable indicator.

Figure 37 shows the RMS error for the two neural networks. The RMS error values share some of the same characteristics as the maximum error values-notably the consistency of the 6-40-1 network's error values at approximately 6%. The 6-20-1 network displayed a larger variation range, but also has significantly lower RMS errors for the 300 and 500 IOP cases than the 6-40-1 network. The 6-20-1 network with 500 IOP's had the lowest RMS error value of 3.0%, and the 6-20-1 network also had the lowest average RMS error value; 5.6% as compared to 6.0% for the 6-40-1 network.

Figure 38 shows the maximum errors obtained when the neural networks were trained to E=0.01. As was seen in Figure 36, the trend in maximum error was towards a reduction as the number of training pairs was increased. Unlike the results shown in Figure 36, there is a much larger maximum error value with the 6-40-1 network and the 50 IOP training set than was seen previously. The only difference between the two 6-40-1 networks with 50 IOP training sets was the training termination error; the reduction in error was effectively modifying the design space mapping in such a way that a poorer representation was being formed in a region where one of the non-training designs was located.

As was noted previously, the 6-20-1 network had a lowest maximum error value with 500 IOPs, but with the change in training error, the maximum error value had risen from 11% to nearly 14%. This indicated that enforcing a tighter training error tolerance can adversely affect the design space representations. The average maximum error values reflected this as well. For E=0.02, the maximum error values were 18.7% and 15.3% for the 6-20-1 and 6-40-1 networks respectively. For E=0.01, these values climb to 21.7% and 18.9% respectively. Similar reductions in accuracy have been noted by Berke and

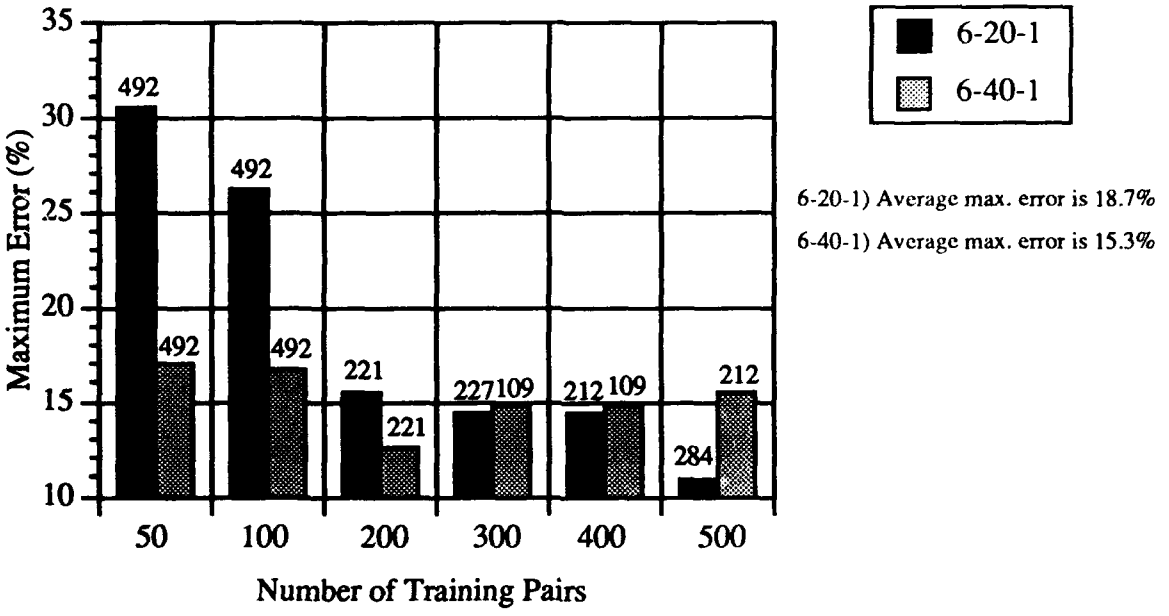


Figure 36. Maximum Error Values (E=0.02) from a Comparison with 500 New Ten Bar Truss Designs

Table 12. Designs Which Resulted in the Maximum Error for the Ten Bar Truss Configuration Design

Design #	X_2	Y_2	X_4	Y_4	X_5	Y_5
492	9.688	4.910	19.641	7.294	9.042	8.052
221	9.030	3.780	20.398	12.523	9.072	7.662
227	10.683	4.290	17.563	9.725	10.880	7.242
109	10.280	4.880	20.353	9.372	9.237	7.474
212	10.153	4.710	22.071	10.558	10.633	7.624
284	10.418	4.120	18.793	10.258	7.429	8.029

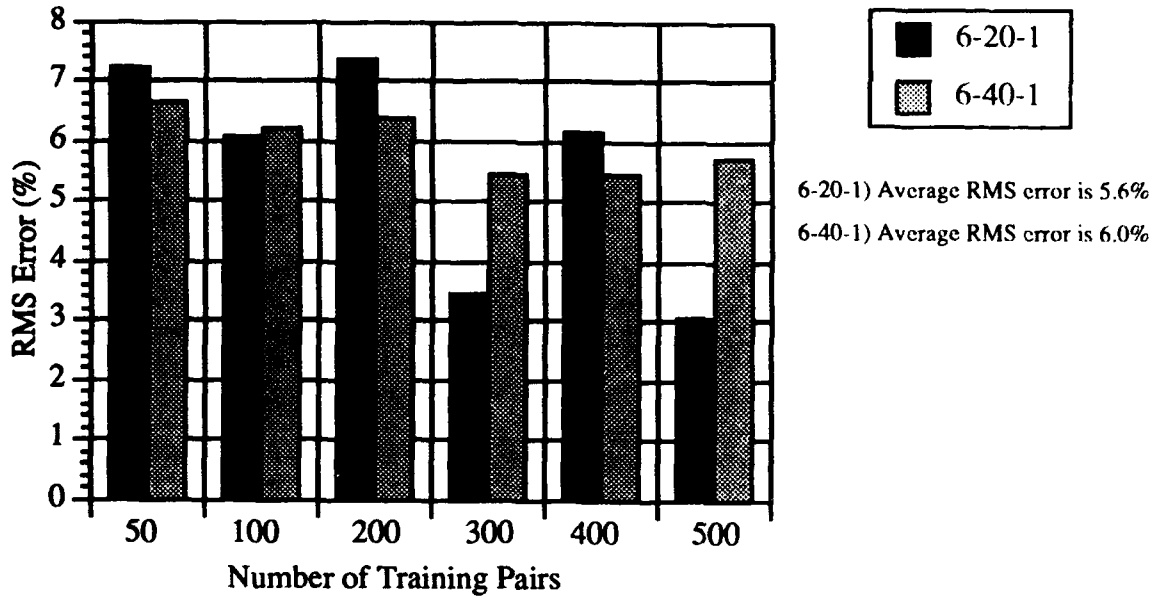


Figure 37. RMS Error (E = 0.02) from a Comparison with 500 New Ten Bar Truss Designs

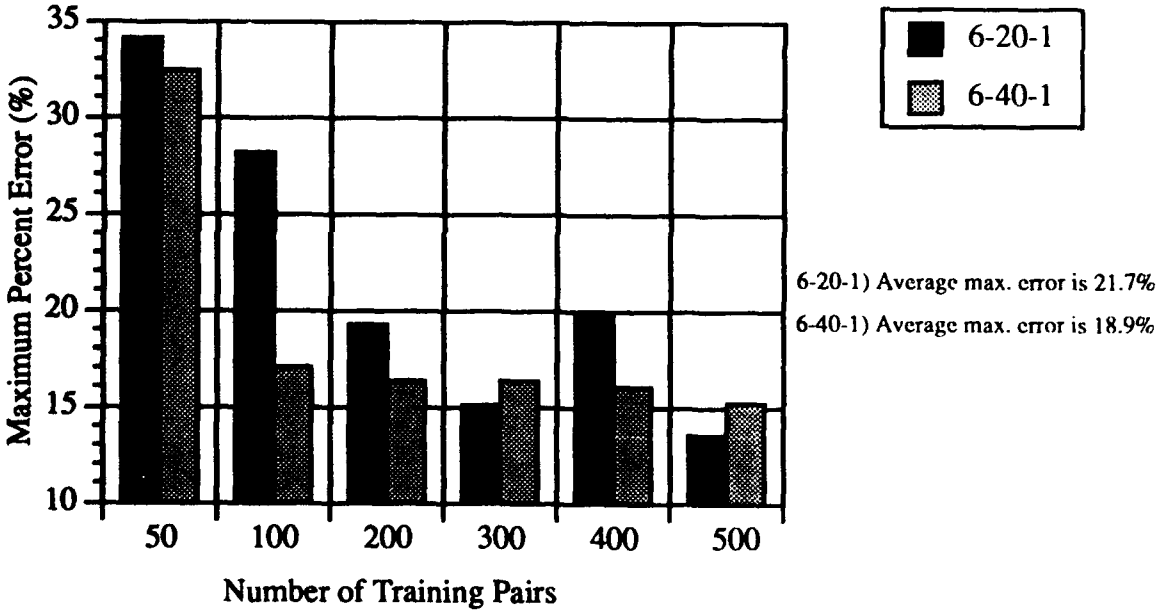


Figure 38. Maximum Error Values (E=0.01) from a Comparison with 500 New Ten Bar Truss Designs

Hajela[11] who noted that a network trained to a higher tolerance (e.g., lower E) matched the training data more accurately (which is exactly what the training process does) but was less accurate for generalization. They defined generalization as the ability of the trained network to accurately predict values in the design space outside of those points in the training set.

The most pronounced change between the two termination error cases was seen for the 6-40-1 network with 50 IOPs in which the maximum error nearly doubled, going from 17% to 32%. All the 6-20-1 networks with E=0.01 showed an increase in maximum error over the E=0.02 cases as well. For the 6-40-1 network increases were seen for the 50, 300, and 400 IOP cases.

Figure 39 shows the RMS error values for the two networks when trained down to E=0.01. The trends in the RMS values were similar to the trends seen for the E=0.02 case of Figure 37 in that there was a reduction in the RMS error values as the IOP number increases. There was however a more significant drop in error from the 100 to 200 IOP case. The RMS error then displayed a lower variation than was seen in the previous case. Also, the 6-40-1 network exhibited a more constant error value, which fluctuated around 6% after 100 IOP. This nearly constant value, in comparison to the 6-20-1 network, was noted for the E=0.01 results. The RMS error values increased over the E=0.02 case, from 5.6% and 6.0% for the 6-20-1 and 6-40-1 networks to 6.3% and 6.4%, respectively, for E=0.01. Also, the lowest RMS error value for E=0.01, 5.6%, was nearly double that of the lowest RMS error value for E=0.02, 3.0%.

The neural networks were interfaced to the optimization algorithm described earlier to solve a general nonlinear programming problem. This allowed the determination of minimum weight truss geometries where weight was considered as the objective function. The neural network representation of the configurational design space was used as the "function" to be minimized (i.e., the weight value predicted for the neural network based on the given truss geometry). Figure 40 shows one such least weight configuration as determined from the neural network/optimization procedure. The FSD weight of this configuration, 6.07 lb, represents a 4% reduction in design weight over the least weight configuration in the training data used. The 6-20-1 network with E= 0.02 and 500 IOP was used to obtain the configurational design shown in Figure 40. This network was chosen because it had the lowest RMS error values for the comparison studies, implying that this network configuration had the best overall representation of the design space. The resultant design obtained from the math-programming procedure was a function of the neural network representation of the design space.

The remaining network/IOP combinations for both the E=0.02 and E=0.01 cases were also used to obtain least weight structural configurations. In every case, node 5 was driven to a (7.00, 7.00 in.) location. There was some difference in the values obtained for nodes 2 and 4, however. Figure 41 shows the (x, y) locations obtained for node 2 from the various networks. The 50 IOP networks had locations of (9.0,-3.0 in.) for node 2. The 100 IOP networks had dissimilar locations for the node location, but a trend was apparent. The remaining designs were clustered in the range of (11,-2.11 in.) to (11, -1.34 in.). The one variation from this trend was for the 500 IOP E=0.02 case, which had a node 2 location at (10.55,-1.54 in.).

Figure 42 shows the locations of node 4 that were obtained from the optimizations. The nodal locations are relatively well clustered, with the 300 IOP case for E=0.01 being the most distant from the clustered region. All designs had node 2 and node 4 locations that were situated in the same "quadrant" for their respective regions.

Figure 43 shows the node 2 locations for the optimal designs obtained from the 6-40-1 networks. When compared to the results for the 6-20-1 network, shown in Figure 41, it is seen that there was less consistency than for the results obtained for the previous network geometry. For the higher IOP networks there was a distinct clustering around (11.0,-1.4 in.). This region was also isolated in the results for the 6-20-1 networks. Unlike

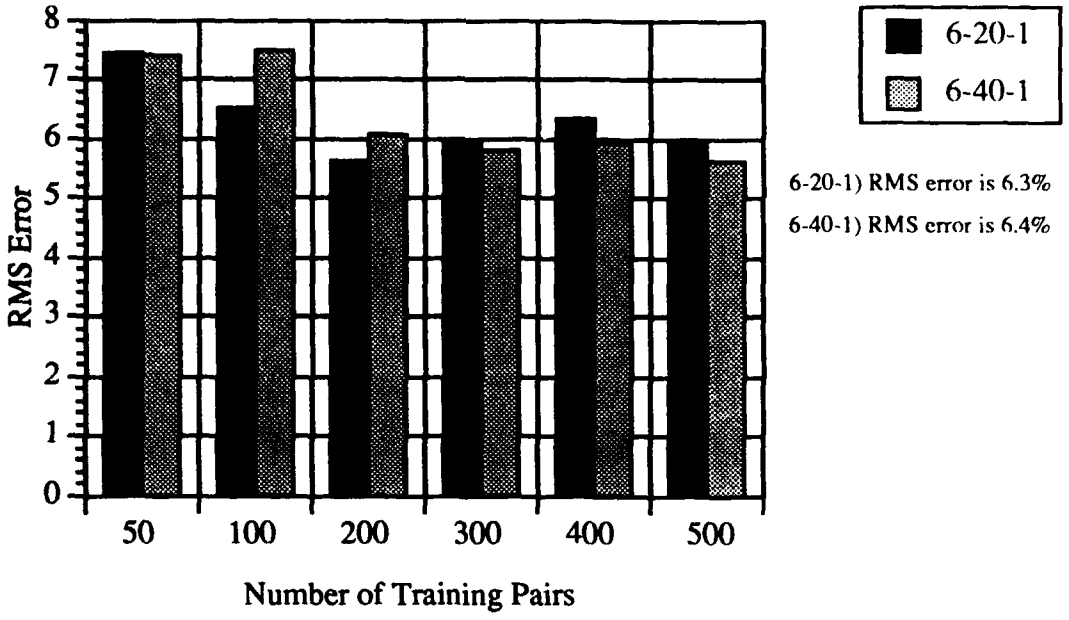


Figure 39. RMS Error (E=0.01) from a Comparison with 500 New Ten Bar Truss Designs

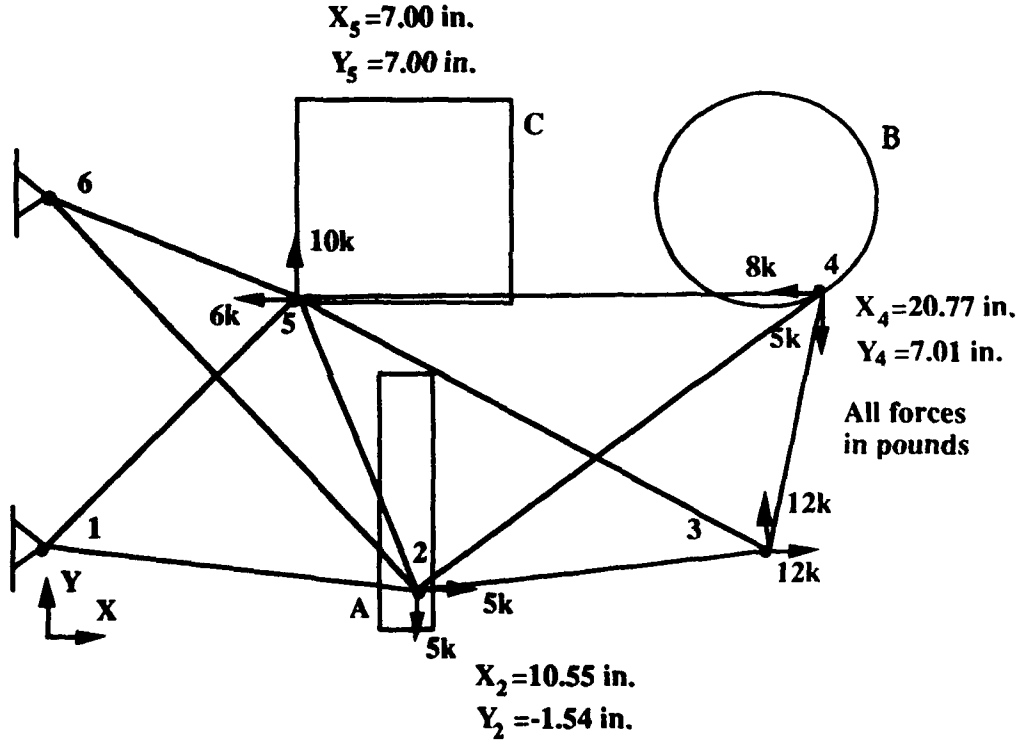
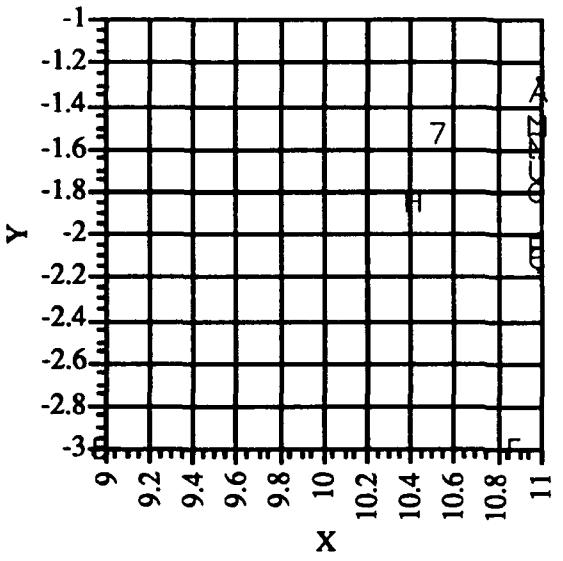
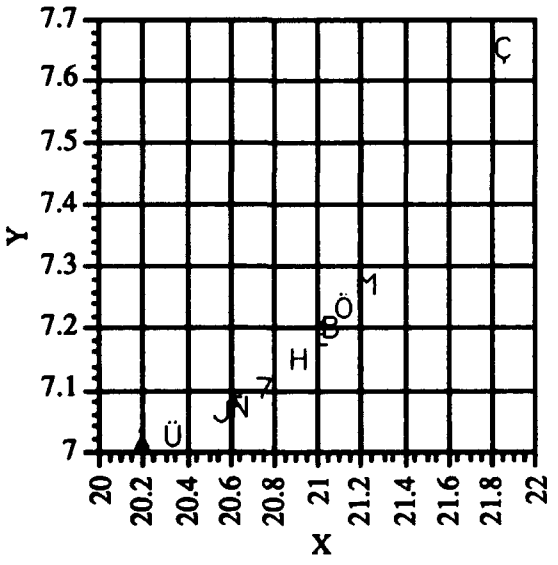


Figure 40. Least Weight Ten Bar Truss Configuration as Determined Using the Neural Network



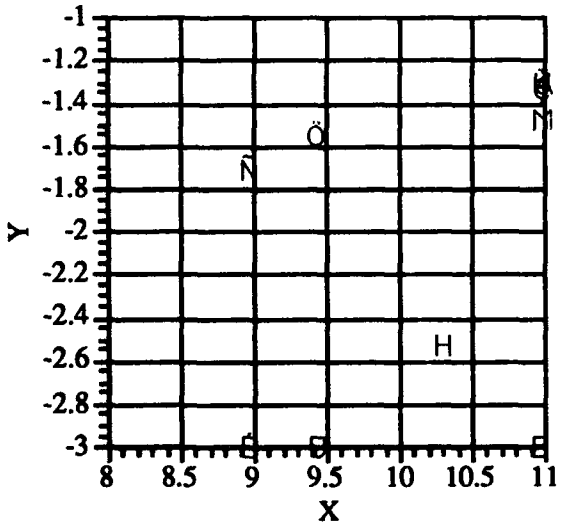
B	50-01 (9.00,-3.00)
J	50-02 (9.00,-3.00)
H	100-01 (10.44,-1.86)
F	100-02 (10.90,-3.00)
Ñ	200-01 (11.00,-1.59)
É	200-02 (11.00,-2.07)
Ç	300-01 (11.00,-2.11)
Å	300-02 (11.00,-1.34)
M	400-01 (11.00, -1.50)
Ü	400-02 (11.00,-1.72)
Ö	500-01 (11.00, -1.82)
7	500-02 (10.55,-1.54)

Figure 41. Node 2 Location for Optimum Designs Obtained Using the 6-20-1 Network



B	50-01 (21.08,7.20)
J	50-02 (20.59,7.06)
H	100-01 (20.94,7.15)
F	100-02 (20.65,7.07)
Ñ	200-01 (20.66,7.07)
É	200-02 (21.05,7.19)
Ç	300-01 (21.87,7.65)
Å	300-02 (20.20,7.01)
M	400-01 (21.24,7.27)
Ü	400-02 (20.35,7.02)
Ö	500-01 (21.14,7.23)
7	500-02(20.77,7.10)

Figure 42. Node 4 Location for Optimum Designs Obtained Using the 6-20-1 Network



B	50-01 (10.99,-3.00)
J	50-02 (9.47,-3.00)
H	100-01 (10.32,-2.54)
F	100-02 (9.46,-3.00)
Ñ	200-01 (9.00,-1.73)
É	200-02 (9.00,-3.00)
Ç	300-01 (11.00,-1.34)
Å	300-02 (11.00,-1.32)
M	400-01 (11.00,-1.49)
Ü	400-02 (11.00,-1.32)
Ö	500-01 (9.45,-1.56)
7	500-02 (11.00,-1.37)

Figure 43. Node 2 Location for Optimum Designs Obtained Using the 6-40-1 Network

the 6-20-1 cases there were four (rather than 2) designs along the Y=-3 axis. In fact, the results for the 50, 100, and 200 IOP cases were the results that were not in the clustered area. This was in contrast to the results for the 6-20-1 network in which only the 50 and 100 IOP cases were not in the clustered area (excluding the 500 IOP E=0.02 case).

Figure 44 shows the locations for node 4 obtained from the optimal designs using the 6-40-1 networks. The design locations were in the same range as was seen previously for the 6-20-1 cases. The 300 IOP E=0.01 case had gone from one extreme to another--from (21.87,7.65 in.) for the 6-20-1 network to (20.07,7.00 in.) for the 6-40-1 network. Also, the 200-02 design had moved from (21,05,7.19 in.) to (21.70,7.53 in.), outside of the clustered region.

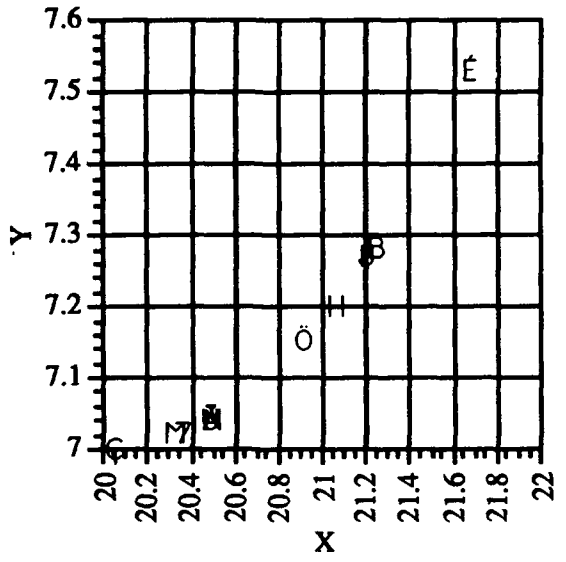
Figure 45 shows a comparison between the neural network predicted weights for the optimal configurations as compared to the actual weights obtained from a FSD re-design of the corresponding configuration. Again, note the improvement in correspondence of design weight prediction as the number of designs used for training was increased. The variations in weight between the levels of network training error were less distinct, although it can be seen that in every case but the 300 IOP case the E=0.02 networks provided a network-predicted weight that was lower than the E=0.01 predicted weight. This does not translate directly to the actual weight trends, however. It can be seen that the actual weights for the E=0.01 versus E=0.02 cases are higher for five of the six cases, the exception being the 300 IOP case (where the E=0.02 design has a lower weight than the E=0.01 design).

Figure 46 shows the percent error between the network predicted weights for the optimal truss configurations and the actual FSD weights for these configurations. The agreement between the network and actual values was seen to improve as the size of the IOP set increased. The error between the predictions for the E=0.01 cases was lower than the E=0.02 cases except for the 300 IOP case, corresponding to the results shown in Figure 45 for this case. The best correspondence was for the E=0.01, 400 IOP case, with an error of less than 1%. The reduction in error between 50 and 100 IOPs was dramatic--the 100 IOP case has less than half the error of the 50 IOP case. The reductions in error beyond the 100 IOP set were more gradual.

Figure 47 shows the same type of comparison as Figure 45, but this figure refers to the 6-40-1 network group. The same trends as seen in the previous figure were again noted, namely that the comparison between the network values and the actual optimum values improved as the amount of training data increased. As was also seen in the previous figure, the network predicted values were typically lower than the actual FSD values. This occurred for all but two cases, 300 IOP and 400 IOP, both with E=0.02. There was more of a variation in the results for the 6-40-1 network for the higher IOP designs (300-500 IOP) as compared to the previous network. The 300 IOP network with E=0.02 provided the lowest weight design, 6.04 lb, as compared to the 400 IOP E=0.02 case for the 6-20-1 network with a weight of 6.05 lb.

Figure 48 shows the percent error between the neural network predicted values and actual FSD values. It was seen that the error level dropped as the IOP number increased. Also note the percent error was typically higher for the E=0.02 networks, the exceptions to this being for the 300 and 400 IOP cases. The lowest error occurred for the 400 IOP E= 0.02 case--less than 0.5%.

The ten bar truss configurational design example was formulated to provide a relatively simple example that described most of the major concepts and implementation issues which arose in the use of neural networks for design space representation. The issues relating to the interpretation of the design space representation were also addressed with this example. The following example, that of a three spar wing-box, was meant to provide a more complex structural model and design task. The same issues that have been discussed for the ten bar truss will again be revisited for this problem; namely, the effective formulation of a design space representation and interpreting the results obtained from the neural network design space representation.



B	50-01 (21.27,7.28)
J	50-02 (21.24,7.27)
H	100-01 (21.08,7.20)
F	100-02 (21.23,7.27)
Ñ	200-01 (20.51,7.04)
É	200-02 (21.70,7.53)
Ç	300-01 (20.07,7.00)
Å	300-02 (20.50,7.04)
M	400-01 (20.33,7.02)
Ü	400-02 (20.50,7.04)
Ö	500-01 (20.94,7.15)
7	500-02 (20.38,7.02)

Figure 44. Node 4 Location for Optimum Designs Obtained Using the 6-40-1 Network

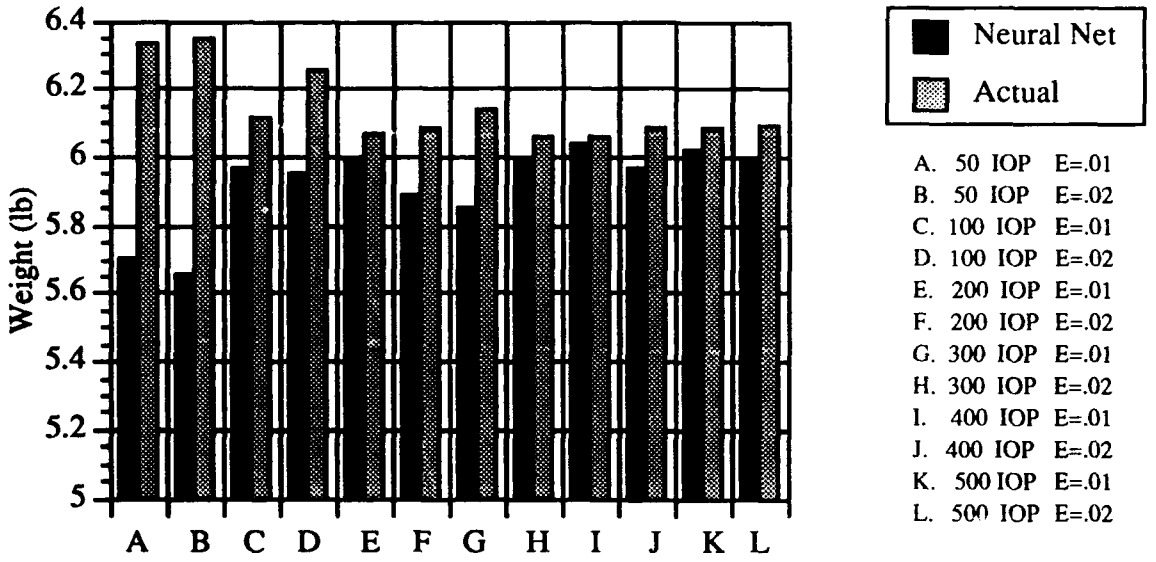


Figure 45. Comparison between Network Predicted Optimum Weights and Actual Optimum Weights (6-20-1 Network)

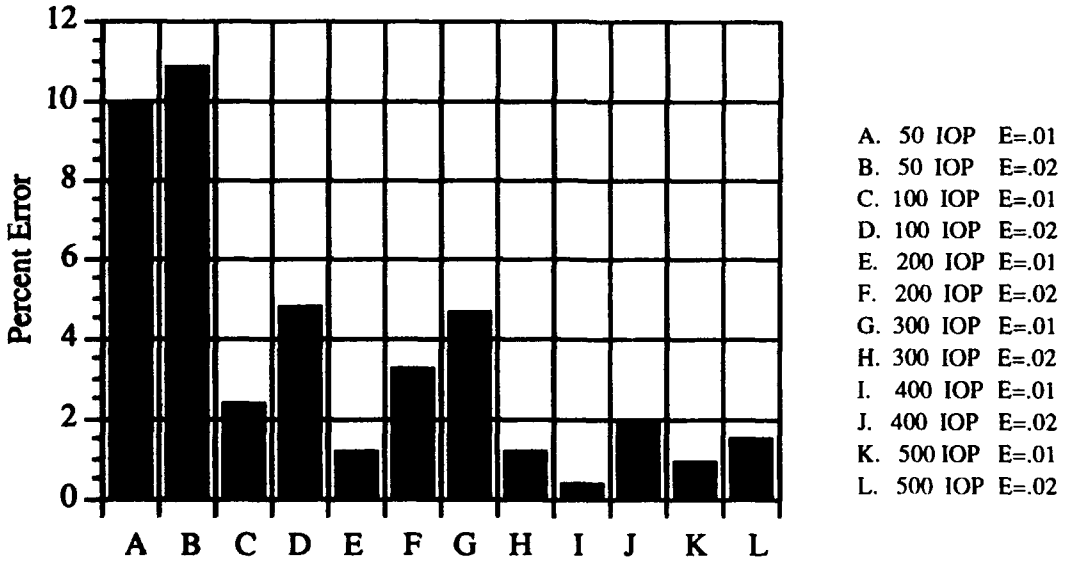


Figure 46. Percent Error Between the Network Predicted Optimum Weights and Actual Optimum Weights (6-20-1 Network)

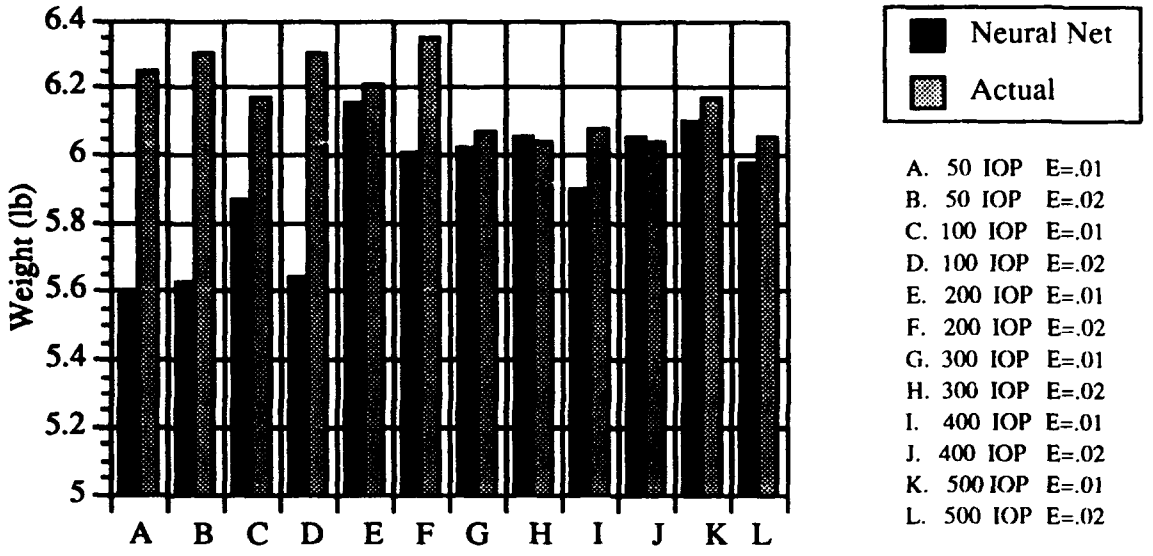


Figure 47. Comparison between Network Predicted Optimum Weights and Actual Optimum Weights (6-40-1 Network)

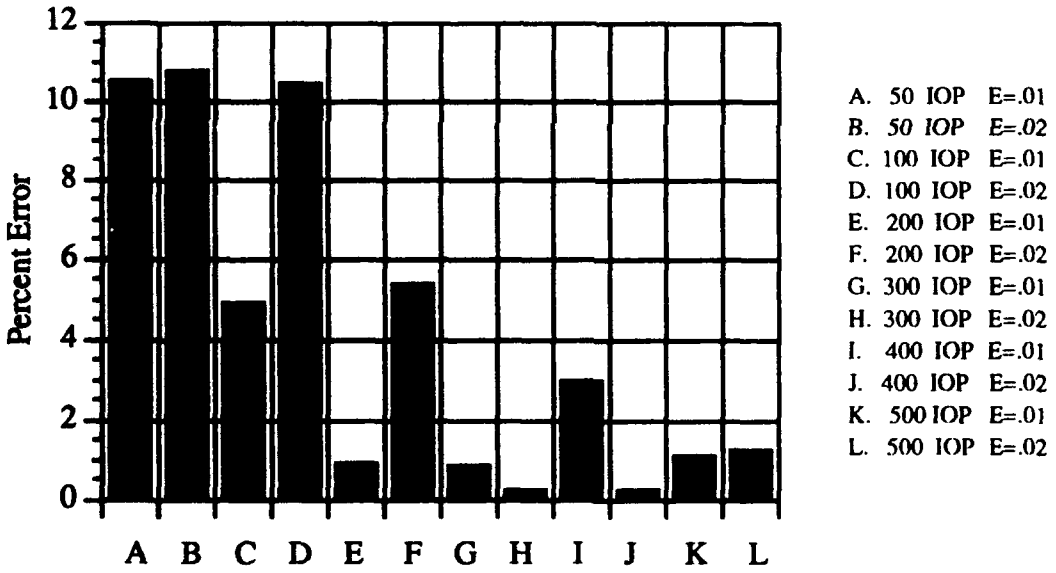


Figure 48. Percent Error Between the Network Predicted Optimum Weights and Actual Optimum Weights (6-40-1 Network)

7.1.2 Three Spar Wing-Box

Spar locations were considered as the configurational design variables for this example. To provide training data for the neural network, sets of training data composed of 20, 50, and 100 different wings were considered. Spar locations were selected at random for the training data. The wings were designed using the FSD approach described previously, and a modal analysis was performed for each of the resulting designs. Three neural networks were used for this example. Each had 3 inputs (the 3 spar locations), a single hidden layer composed of 20, 40, or 80 nodes, and an output layer composed of 5 neurons. The five outputs were FSD weight, tip deflection at the main spar location, first, second, and third natural frequency. The networks were trained to an error tolerance of E=0.02. A second set of 100 wing configurations were designed, and then propagated through the neural networks for comparison purposes. This provided a means to determine the design space representation "quality" of the various hidden layer and training pair combinations.

Table 13 illustrates the fully-stressed design procedure's ability to consistently identify least weight designs. The fully-stressed design procedure was used to design a wing with spar locations at 25, 45, and 65% chord. Twenty designs were produced, each design starting from a different set of randomly selected element gauges. As can be noted from the table the tolerance and RMS error values are low. The highest tolerance and RMS errors occurred for the second natural frequency.

Figure 49 shows the maximum errors that were found when the 100 new designs were propagated through the 3-20-5 networks. Two numbers are shown in the legend; the first is the number of neurons in the hidden layer, and the second is the number of training pairs in the IOP set. The expected trend in the error, which was noted for the ten bar truss example, was the reduction in error as the number of training pairs was increased. All quantities but the tip displacement errors showed this trend. There was a small increase in error from the 50 to 100 IOP case for tip displacement. Most error values were below 1%, but the second natural frequency had a maximum error value that was greater than 5% for the 20 IOP case. The third natural frequency had a 2% error for the 20 IOP case. These relatively high errors were due to the relatively high range of values that the second and third natural frequency assumed as compared to the other three network outputs. The second natural frequency ranged from 14.4 to 22.4 Hz, a 56% increase. The third natural frequency ranged from 30.3 to 33.6 Hz, an 11% increase. The first three network outputs had 11%, 20%, and 22% variations, respectively (which were much lower than the second natural frequency value). As the IOP number increased from 20 to 50, there was a dramatic reduction in error for the second and third natural frequencies.

Figure 50 shows the maximum errors for the 3-40-5 networks. Again, as expected, the most prominent trend was the reduction in error as the IOP number increased. As might be expected from the previous figure, the second natural frequency had the highest error, this time nearly 7% for the 20 IOP case. The first natural frequency error values did not follow the expected trend. The 100 IOP error was larger than either the 20 or 50 IOP case.

Figure 51 shows the maximum error values for the 3-80-5 networks. The expected trend towards a reduction in error for increasing IOP number is again illustrated. As compared to the previous two cases, the 20 IOP tip displacement showed a steep increase in error (over six times as large as the 3-20-5 network and nearly double the 3-40-5 network). All output quantities showed the expected reduction in error as IOP number increased.

Three different networks were considered so that the effect that the hidden layer size had on design space representation could be noted. Rather than consider individual network output quantities for this comparison, the maximum errors were averaged for the

Table 13. Uncertainty in the Three Spar Wing Design FSD Results

Value	Tip Disp. (in.)	Weight (lb)	1st Natural Freq. (HZ)	2nd Natural Freq. (HZ)	3rd Natural Freq. (HZ)
Maximum	11.8462	265.9119	7.4879	17.4193	32.6931
Minimum	11.7910	265.1795	7.4699	15.9741	32.6350
Average	11.8186	265.5457	7.4789	16.6967	32.6641
Tolerance	± 0.0276	± 0.3662	± 0.0090	± 0.7226	± 0.0290
RMS Error (%)	$1.469(10^{-2})$	0.2272	$4.699(10^{-3})$	0.4470	$1.895(10^{-2})$

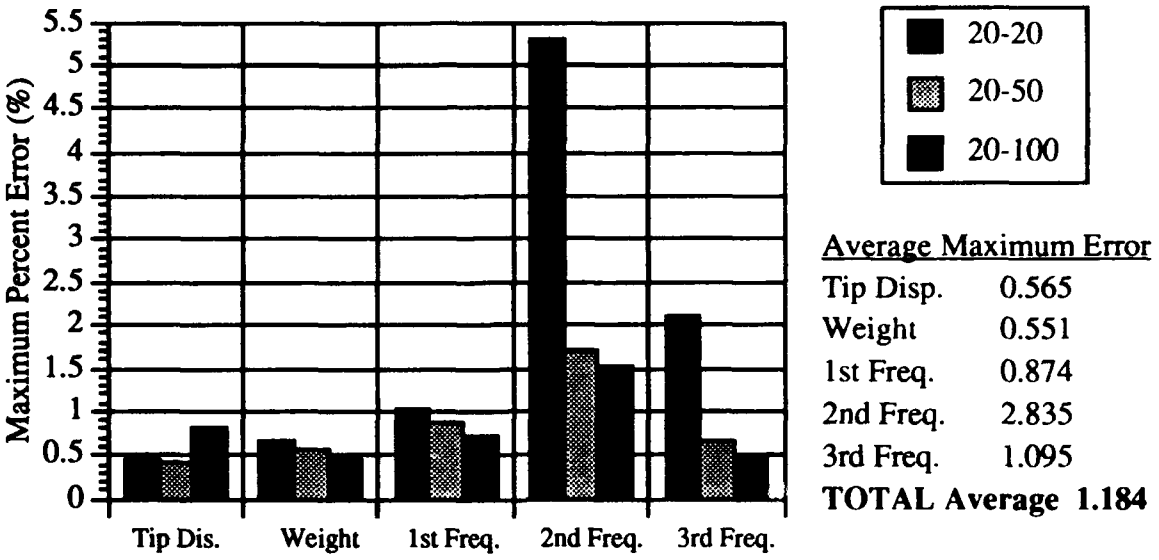


Figure 49. Comparison of Maximum Errors for the Three Spar Wing (3-20-5 Network)

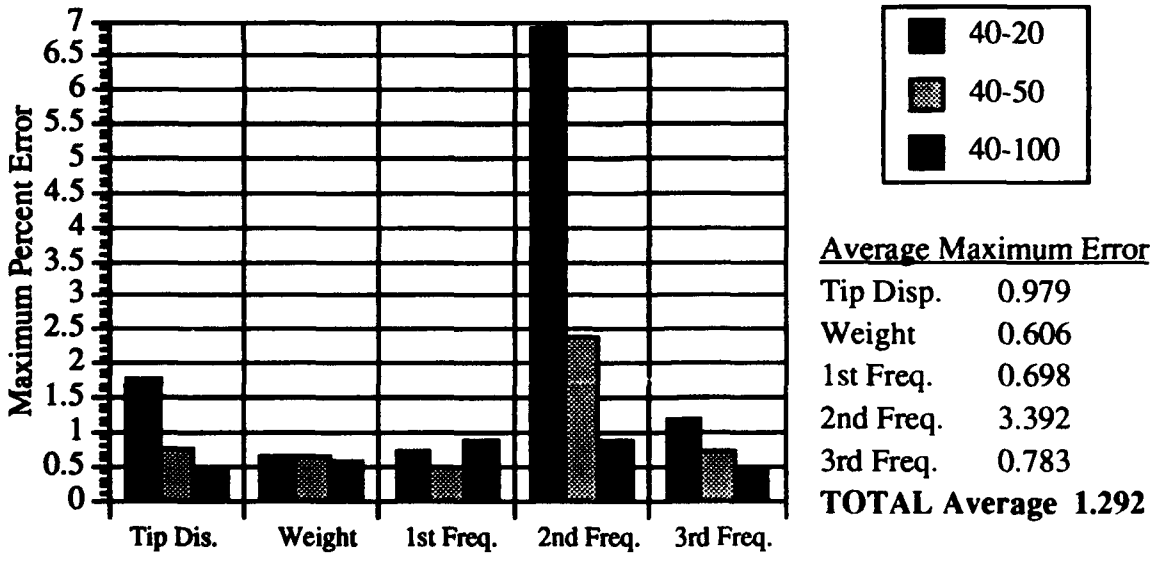


Figure 50. Comparison of Maximum Errors for the Three Spar Wing (3-40-5 Network)

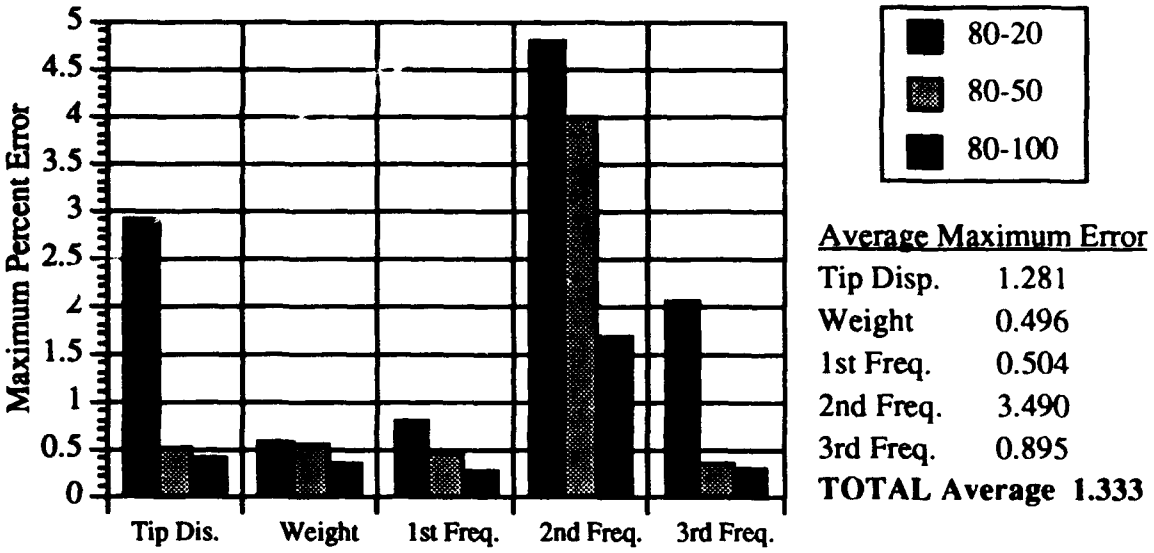


Figure 51. Comparison of Maximum Errors for the Three Spar Wing (3-80-5 Network)

comparison. The 3-20-5 network provided the lowest average maximum error value, 1.184%, followed by the 3-40-5 network with a value of 1.292%. The 3-80-5 network had the highest average maximum error value, 1.333%. Increasing the number of neurons in the hidden layer can be associated with an increase in the average maximum error value for this particular example.

Figure 52 shows the RMS errors for the 3-20-5 networks. The expected trends in error were easily identified--a reduction in error as the IOP number increased. As might be expected, the second natural frequency had the highest RMS error value. The RMS error for the weight (20 IOP case) was also relatively high. This was unexpected from the maximum error data, but it implied that more of the 100 new designs were less accurately represented by the neural network for the 20 IOP case.

Figure 53 shows the RMS errors for the 3-40-5 networks. Reduction in error as the IOP number increased is readily apparent. The second natural frequency again had the highest RMS error values, but the tip displacement error for the 20 IOP case was much more comparable to the other RMS errors for the network outputs. The third natural frequency errors have been reduced over those of the 3-20-5 cases.

The RMS errors for the 3-80-5 networks are shown in Figure 54. The weight error for the 20 IOP case had risen appreciably, nearly double that of the 3-40-5 network. The second natural frequency shows the same large error as well. The tip displacement error for the 20 IOP case was also appreciably higher than the corresponding case for the previous two networks.

In order to make a generalized comparison of the quality of the three networks for design space representation, the RMS errors were averaged for each of the three networks. The 3-20-5 networks had the highest average RMS error, with a value of 0.71%. This was followed by the 3-80-5 networks, with an average RMS error of 0.42%. The 3-40-5 networks had the lowest RMS error, with a value of 0.41%. These results were different from the trends noted for the average maximum error, which indicated that an increase in the number of neurons in the hidden layer corresponded to an increase in the average maximum error. The RMS results almost indicate an opposite trend. The 3-20-5 networks had a significantly higher value than either the 3-40-5 or 3-80-5 networks (both of which had nearly identical average error values). This could imply that there was an optimal network geometry somewhere in the 20-80 neuron range for the hidden layer (for this particular problem).

The neural network representations of the design space for the three spar wing-boxes were interfaced to the math-programming optimization procedure to extract a set of optimal configurations. Each of the network output quantities (tip displacement, weight, first, second, and third natural frequency) were used independently as the objective functions for the optimization. Table 14 shows a comparison between the neural network predictions for this set of optimal design values and the actual FSD values for wings of corresponding geometry. It can be seen from the table that there is a good correspondence between the values, showing that the neural network has in fact represented the design space accurately. The 40 neuron hidden layer network trained with 100 IOP was used since this network exhibited the lowest average RMS error for the comparative study performed earlier.

The minimum weight design was a "narrow" wing-box. The leading and trailing edge spars were positioned at their upper and lower constraints, respectively. The leading edge spar, at 30% C, was at the maximum thickness location for the NACA 4412 airfoil. The wing-box cross section is effectively "tall and thin," maximizing the second moment of inertia.

The minimum tip deflection design had both the leading and trailing edge spars at their most forward and aft positions, respectively. In this case the wing-box was "wide and thin." This wing was much heavier than in the previous case due to the addition of material--material that provided a lower displacement value. Not surprisingly, the

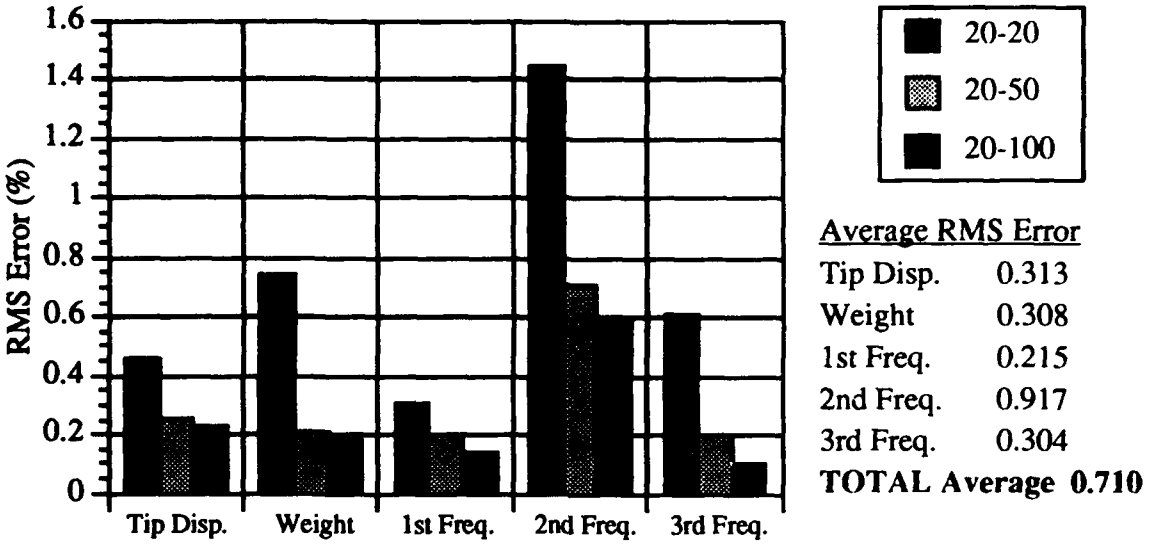


Figure 52. RMS Errors for the Three Spar Wing (3-20-5 Network)

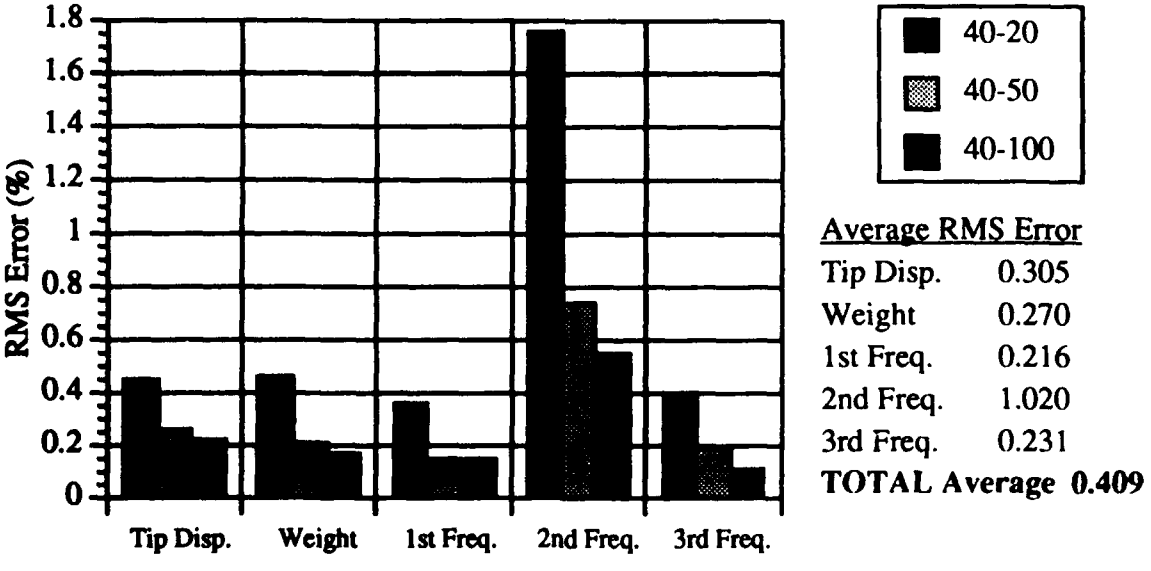


Figure 53. RMS Errors for the Three Spar Wing (3-40-5 Network)

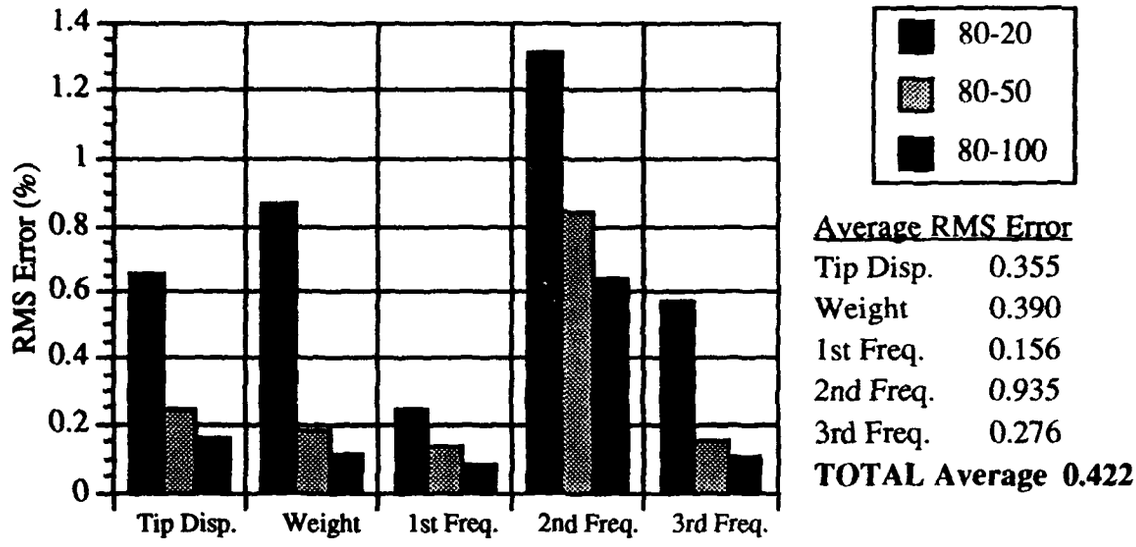


Figure 54. RMS Errors for the Three Spar Wing (3-80-5 Network)

Table 14. Optimum Design Results for Three Spar Wing-Box Configuration Design

DESIGN OBECTIVE	NN OPTIMAL SPAR LOCATIONS (% Chord)	δ_{NN} (in.)	δ_{FSD} (in.)	W_{NN} (lb)	W_{FSD} (lb)	ω_{1NN} (Hz)	ω_{1FSD} (Hz)	ω_{2NN} (Hz)	ω_{2FSD} (Hz)	ω_{3NN} (Hz)	ω_{3FSD} (Hz)
W_{MIN}	30.0, 38.0, 60.0	12.19	11.77	249.1	246.9	8.00	8.16	13.63	12.78	33.43	33.66
δ_{TMIN}	15.0, 35.0, 75.0	11.32	11.38	304.0	305.3	6.68	6.67	20.82	21.00	30.45	30.50
ω_{1MIN}	15.0, 55.0, 75.0	12.62	12.56	306.3	311.6	6.41	6.33	23.14	22.99	29.94	29.57
ω_{2MIN}	30.0, 35.0, 60.0	12.25	11.82	249.3	246.8	8.01	8.16	13.58	13.11	33.20	33.56
ω_{3MIN}	15.0, 55.0, 75.0	12.62	12.56	306.3	311.6	6.41	6.33	23.14	22.99	29.94	29.57

dynamic characteristics of this design were quite different from the minimum weight case.

The minimum first natural frequency design was, effectively, the thinnest wing-box possible. As in the minimum tip deflection case, the leading and trailing edge spars were at their lower and upper bounds, respectively. The main spar was at its most aft position, where its height was a minimum. This mode was a first beam bending mode.

The minimum second natural frequency design was very similar in geometry to the minimum weight design. The leading and trailing edge spars were at their upper and lower bounds respectively, while the main spar was at the position where it had a maximum height. Again, the cross section was narrow and tall. This mode was a torsional mode.

The minimum third natural frequency design was identical to the minimum first natural frequency design. The leading and trailing edge spars were at their lower and upper bounds, respectively, and the main spar was at the position where its height was minimized. This resulted in a wide, thin cross-section for the wing-box. This was a second beam bending mode.

In addition to the above designs, optimal designs based on a "weighted" objective function were also possible. This would have involved formulating the objective function as a weighted function of the five output quantities. This possibility was not explored at this time, but it is feasible to incorporate multiple requirements into the process.

7.2 Configurational Design Using Recursive Learning

Two examples were considered: the configurational design of a ten bar truss for minimum weight and the configurational design of a four spar wing-box where tip displacement, FSD weight, and natural frequency were considered independently as the objective functions. A description of the impact that the recursive learning procedure had on the use of neural networks is presented along with the results obtained for both examples.

7.2.1 Ten Bar Truss

The ten bar truss example employed a 2-80-80-1 network (2 inputs, 80 neurons in the first hidden layer, 80 neurons in the second hidden layer, and 1 output neuron). An initial training set of ten randomly selected designs was used to start the procedure. At each iteration in the recursive procedure two new designs were generated as described previously.

Three of the recursively trained networks which were developed as part of this iterative process are presented in this report: the initial network with 10 IOP, the network trained with 20 IOP, and the network trained with 40 IOP. Graphical presentation of the neural network representation of the design space was accomplished by developing contour plots from data determined by propagating a 100 by 100 set of evenly spaced nodal coordinates through the neural networks at each stage of the design space evolution. This particular problem was selected to allow for the graphical presentation of a two-dimensional design space.

Figure 55 shows the weight trends obtained from the baseline neural network trained with only ten IOP. These initial ten IOPs, shown as circles in the figure, were randomly selected (x, y) coordinate locations for node 2. The design space was poorly defined at this early stage; weight appeared to vary in a very predictable manner with changes in geometry. This was due to the limited information provided by only ten designs.

Figure 56 shows the design space after ten additional designs were added. Those designs added randomly are denoted once again by circles. The five designs determined by the recursive procedure as the least weight configurations are represented by the

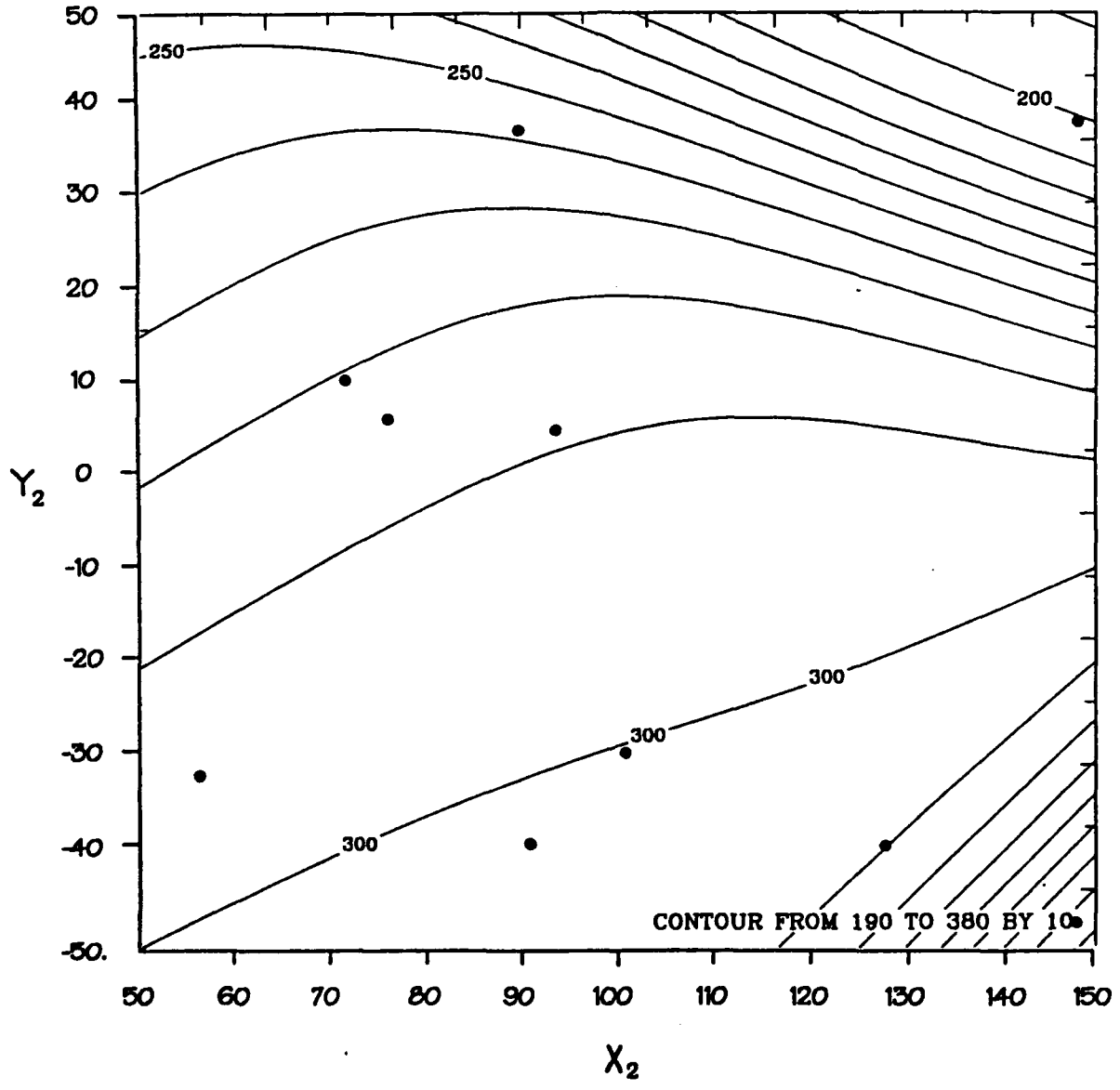


Figure 55. Ten Bar Truss Design Space, 10 IOP

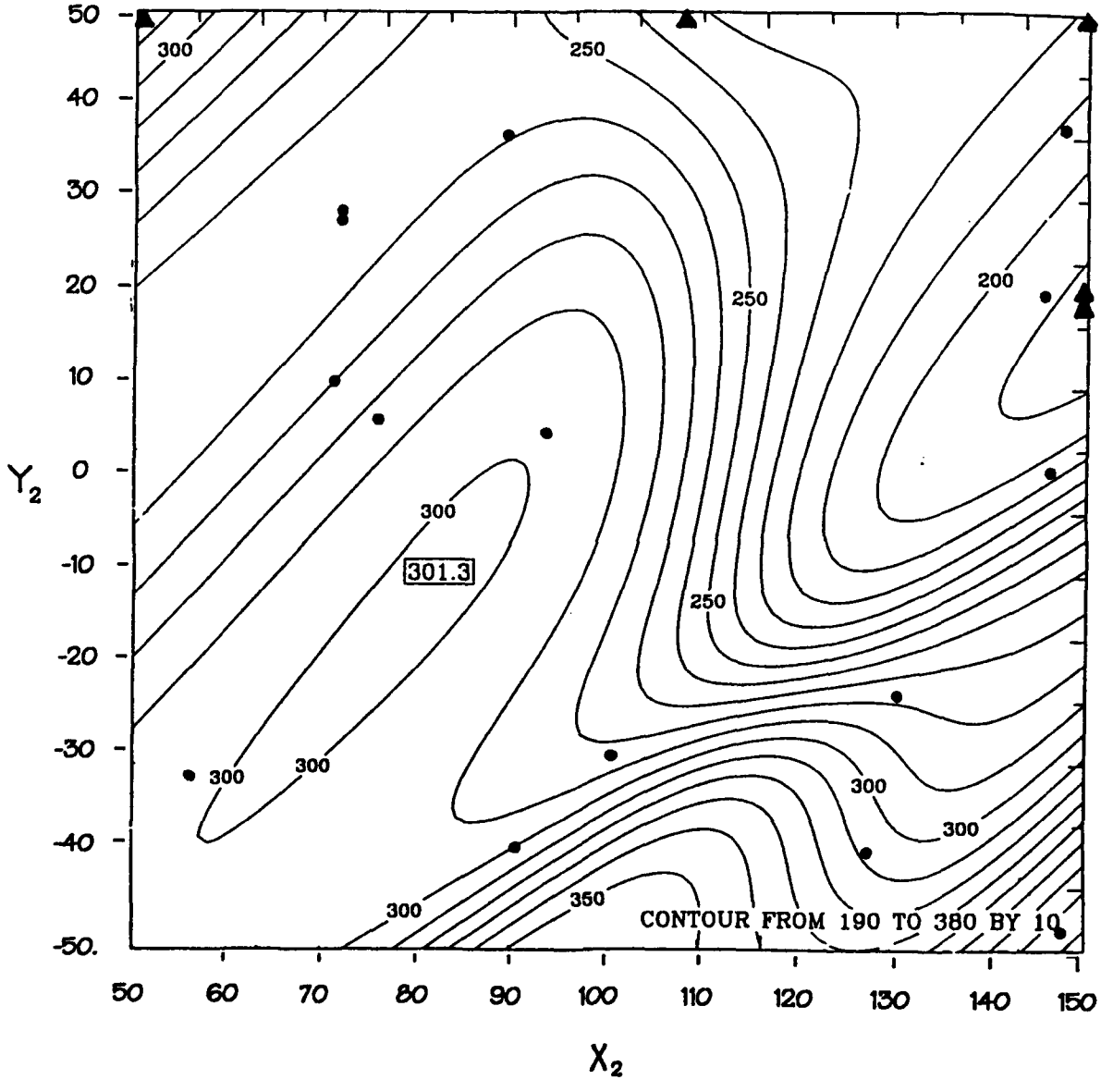


Figure 56. Ten Bar Truss Design Space, 20 IOP (15 Random, 10 Recursive)

triangular symbols. Even at this early stage the recursive procedure has been able to isolate the global minimum for weight (node 2 located near 150., 20. in.). Two of the recursive designs were located in this region. The locations of the other recursive designs are of interest. They appear to represent local minima within the neural network representation at interim steps in the recursive process. These designs resulted from the use of the procedure which determined these points. The gradient based optimization was initiated using a random starting point for the node 2 coordinates. This apparently resulted in the optimization isolating local minima along constraint boundaries. An alternative approach may have been to perform multiple optimizations with a set of random starting locations, and then select additional candidate designs from the resultant set.

The design space representation using 40 designs in the network training is illustrated in Figure 57. Hexagonal symbols represent the ten additional network predicted least weight configurations added to the training data. The randomly selected training data is again represented by a solid circular symbol. Four of the network-predicted optimal designs were at the global minimum. One other, at (133., 10. in.), was also not far from the optimum. Four of the other network predicted optima were clustered about a local minimum at (50., 20. in.), while the final optimal design was at (90., 50. in.). The single random starting location for the optimization procedure was the likely cause for the latter five designs (i.e. local minima). The evolution of detail in the design space is indicated in Figures 55-57. Several local maxima and minima appeared as additional information was added to the training set. It is interesting to note that a local minimum was formed at (120., -10. in.) in a region where no training data was present. This local minimum, shifted to a location of (135., -10. in.) with additional design space detail as seen in Figure 58.

Figure 58 illustrates a design space representation formulated for comparison purposes. It was obtained from a neural network with the same network geometry as above but it was trained using 100 randomly selected designs. The global minimum at (150., 20. in.) is readily apparent, along with the other local minima and maxima that were seen previously. The two predominant local minima seen in Figure 56 are further refined in this figure. It is interesting to note that the recursive neural network was able to describe the local minima at (120., -10. in.) in Figure 57, which is clearly defined in Figure 58, shifted to (135., -10. in.). The neural network was able to isolate this local minimum even though there was no training data in this region of the design space for the 40 IOP case.

Because of the two-dimensional nature of the design task for the ten bar truss, this example has provided a "graphical" means to inspect the design space development as the recursive learning procedure progressed. The procedure was shown to be effective for defining the characteristics of the design space with small sets of training data. Trends in the design space were also identified even though there was no actual training information located within that region of the design space. The limitations of the recursive procedure involve isolating local minima rather than the global minimum. This limitation could be overcome by multiple starting locations for the network-based optimization phase of the procedure or the use of an alternative algorithm for adding data to the training set.

A comparison was performed on the neural network representations by propagating the 100 random designs used to create Figure 58 through each of the three networks. The 10 IOP network had a maximum error value of 22.09% and an RMS error value of 1.43%. The 20 IOP network had a maximum error value of 18.44% and an RMS value of 0.71%. The 40 IOP network had a maximum error value of 18.35% and an RMS error of 0.65%. The expected decrease in maximum and RMS error as the IOP number increases was noted. The 100 IOP network, when compared to its own training data, had a maximum error value of 2.51% and an RMS error value of 0.10%.

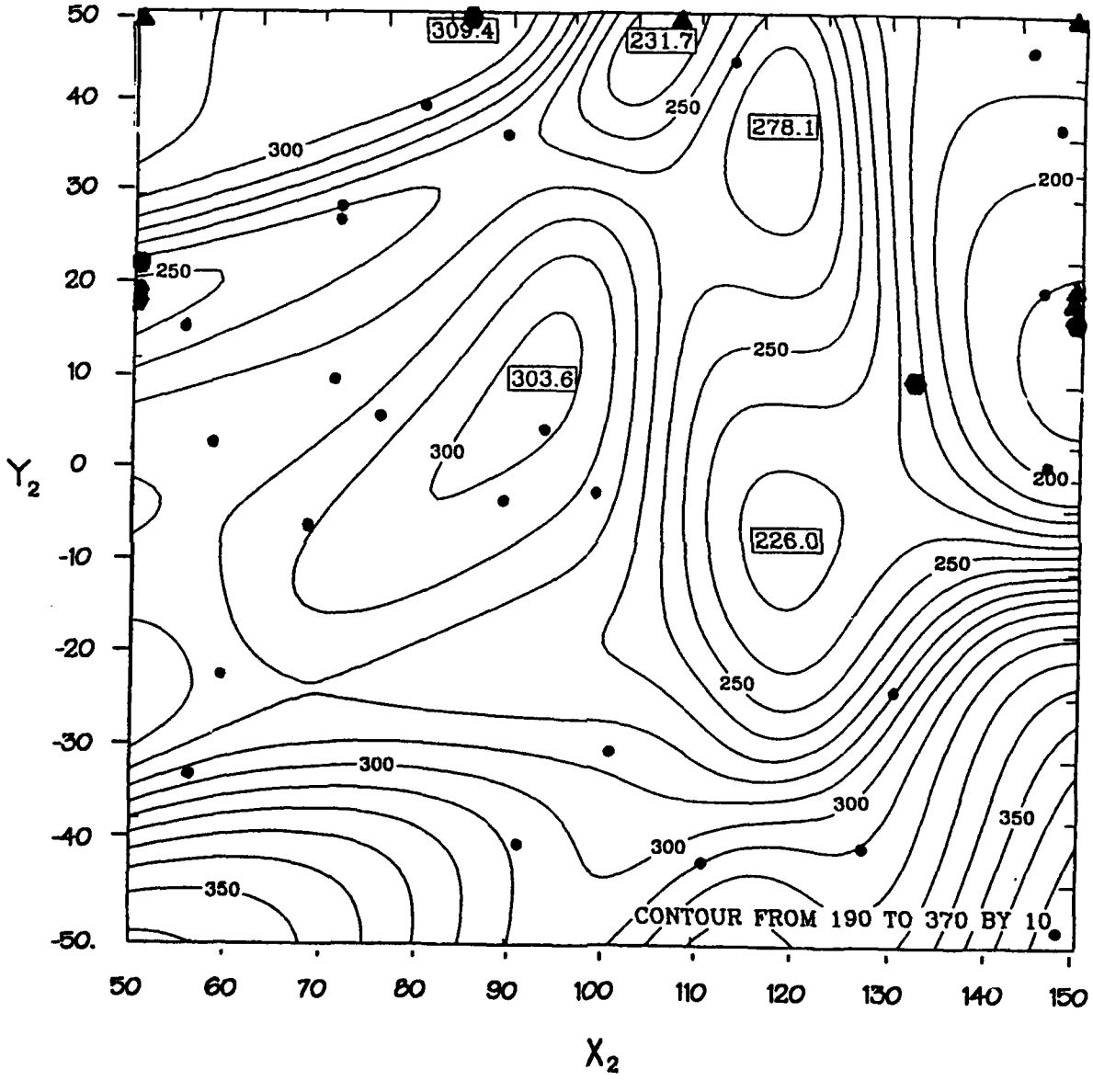


Figure 57. Ten Bar Truss Design Space, 40 IOP (25 Random, 15 Recursive)

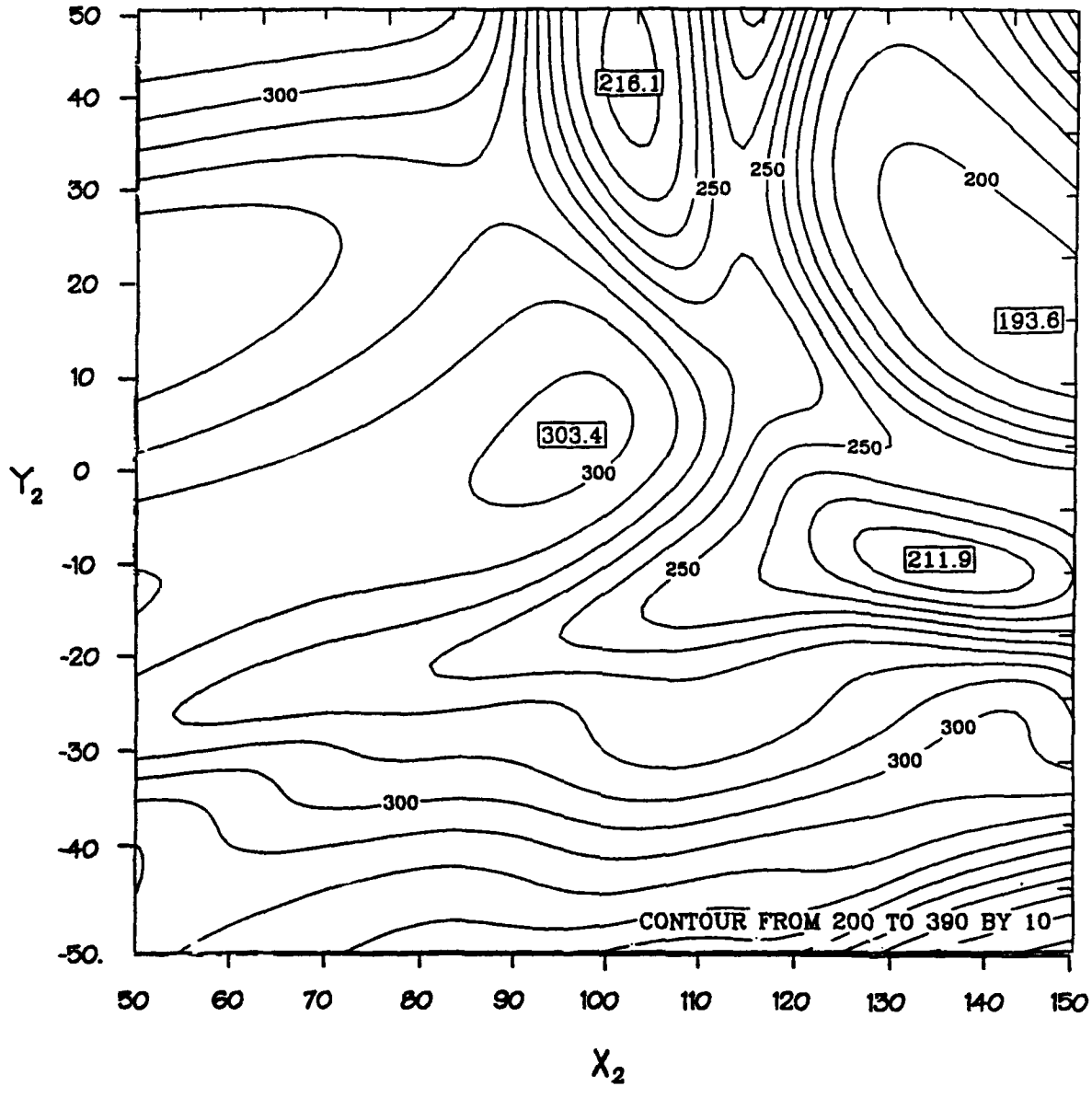


Figure 58. Ten Bar Truss Design Space, 100 IOP Random

For this study termination of the recursive procedure was done based on a decision from the user. Design space refinement was monitored and at some point a decision was made to conclude the recursive process based on the design space representation's response to the recursive training. There is no simple means apparent to the authors for a generalized termination criterion for all design space representations. Thus the termination decision was left to the real, versus artificial, intelligent designer. The number of recursive iterations employed for the examples reflect this decision.

7.2.2 Four Spar Wing-Box

The wing-box chordwise spar locations were considered as the configurational design variables for this example. The neural network used for this study had 4 inputs (the 4 spar locations), a single hidden layer composed of 40 nodes, and an output layer composed of 5 neurons. The five outputs were the design weight (as determined from the FSD procedure for the single static load), the corresponding tip deflection under static load, and the first, second, and third natural frequencies.

Since the goal of the study was not to develop a single point design but to provide a means for representing the concept's design space, a recursive procedure was required which would help provide detail in the more critical regions of the multi-dimensional design space. Also since the design space cannot be "visualized" in as straightforward a manner as the previous example, the data will be presented in summary table form. Previous efforts have shown that the actual output from the neural network was not particularly accurate but does provide good "trend" information. For this reason all the values shown in the following tables are the results obtained from the actual FSD designs of the neural network predicted optimal geometry wing configurations.

The network was trained with a baseline set of data composed of only two designs. Table 15 shows the two baseline optimum designs, the spar locations, weight, tip displacements and natural frequencies. These two designs were selected at random from a subset of 32 designs uniformly distributed over the allowable design space. Interestingly, this random selection process lead to only a variation in the 4th spar as indicated in the table.

The five network outputs were then used in a cyclic fashion as independent objective functions to drive the recursive training. Tip displacement was considered as the first objective function, followed by weight and then the three natural frequencies. The subsequent optimization provided a new candidate configuration which was "re-designed" using the FSD procedure and added to the training set. After the third natural frequency was used, the weight again became the objective function. Twenty-five iterations were performed, so that each network output was used as the objective function five times. Each of the 25 optimal design predictions obtained from the neural network was used as additional training data to further refine the design space. A comparison between the recursively predicted optimum values and two other network predictions was also performed. In the first case 27 randomly selected IOP were used. This was the same size as the training set developed using the recursive procedure. In the second case 100 randomly selected designs were used in the training set.

The iteration history for those designs that considered weight as the objective function is shown in Table 16. Interestingly, the design selected after the second iteration, which resulted from the network trained with only "three" designs, forced each spar to a constraint limit and resulted in what appears to be a near optimum weight design. The first, third, and fourth spar locations were the same for all the designs and were located at constraint boundaries. The second spar location showed a slight variation in location as additional designs were added to the training set. The weight was reduced from 270.6 lbs for the first iteration to 270.1 lb for the fifth iteration. The displacement and natural frequencies also appear relatively insensitive to changes in the second spar location. The optimal weight configuration obtained from the network trained with 100 random designs

Table 15. Four Spar Recursive Configuration Design, Baseline Designs

SPAR LOCATIONS (% Chord)	W (lb)	d (in.)	ω1 (Hz)	ω2 (Hz)	ω3 (Hz)
22.5, 40., 55., 67.5	288.1	7.99	7.01	17.32	32.23
22.5., 40., 55., 72.5	299.1	7.92	6.82	18.81	31.70

Table 16. Four Spar Recursive Configuration Design, Weight Based Designs

DESIGN OBJECTIVE	NN OPTIMAL SPAR LOCATIONS (% Chord)	W (lb)	d (in.)	ω1 (Hz)	ω2 (Hz)	ω3 (Hz)
WMIN	2) 30.0, 45.0, 50.0,65.0	**270.6**	8.22	7.43	14.56	32.55
	7) 30.0, 36.2, 50.0, 65.0	**270.2**	8.11	7.45	14.57	32.82
	12) 30.0, 40.3, 50.0, 65.0	**270.2**	8.11	7.46	14.33	32.80
	17) 30.0, 40.1, 50.0, 65.0	**270.1**	8.10	7.46	14.30	32.82
	22) 30.0, 37.5, 50.0, 65.0	**270.1**	8.10	7.45	14.47	32.82
27-Random	20.4, 36.8, 52.1, 65.0	**285.9**	8.00	6.99	17.13	32.53
100-Random	20.8, 38.0, 50.0, 65.4	**286.8**	7.98	7.01	16.98	32.69

was rather poor in terms of the objective function value, when compared to the recursive results. This weight was 286.8 lb, 6.2% higher than the 270.1 lb recursive value.

Table 17 presents the results obtained for the iterations where tip displacement was considered as the objective function. It can be seen that there was only a slight improvement in the objective function as the iterations progressed; a drop from 7.83 in. to 7.81 in. The third and fourth spar locations remained fixed throughout, with the first and second spars showing some variation in location. Of these two, the second spar showed a relatively small variation in position (35-38.6%C) as compared to the first spar (15-22.4%C). The variations in the displacement, weight, and natural frequency were significantly higher than were seen when weight was used as the objective function. These quantities were more strongly influenced by the location of the first spar than the second. This seems reasonable since the first and fourth spar locations effectively defined the width of the wing-box cross-section, while spars 2 and 3 influenced the thickness characteristics of the wing-box. The designs obtained from the network trained with either 27 or 100 random designs showed higher values when compared to the recursive case. Recall that because the neural network training data involved fully-stressed designs that each of the designs shown in the tables are least weight, FSD designs.

The designs obtained when the first natural frequency was considered as the objective function are given in Table 18. As the iterations progressed, a drop in the first natural frequency from 6.49 Hz to 6.33 Hz was obtained. The first and fourth spars remained fixed in location for each of the five iterations based upon this frequency. The second spar changed location only once from 35% to 45% C between iterations 1 and 2. The third spar location showed a variation in position from 53.1% to 60.0%. The displacement, weight, and natural frequency characteristics seem relatively insensitive to changes in third spar location, further enforcing the presumption made earlier that the first and fourth spar locations were predominant in defining the design characteristics. The best design obtained from the networks trained with the 27 and 100 designs yielded a lowest first natural frequency of 6.39 Hz as compared to 6.33 Hz which was obtained recursively.

The designs obtained when the second natural frequency was considered as the objective function for the recursive procedure are shown in Table 19. The minimum second natural frequency dropped from 14.57 Hz to 14.34 Hz as the iterations progressed. Spar 2 showed a variation in position among the five designs, while spars 1, 3, and 4 remained fixed in location (with the exception spar 3 which changed location once from 50% to 40% C). These designs were very similar to the least weight designs in geometry, and again it was seen that the spar 2 location has little effect on the design characteristics. The minimum second natural frequency design obtained from the 100 random IOP network shows a poor value for the second natural frequency, 17.01 Hz as compared to 14.34 Hz obtained using recursive learning. The designs obtained when the third natural frequency was considered as the objective function indicated similar behavior.

The four spar wing example describes the application of recursive learning to a more complex design problem. The characteristics of the design space were identified through this approach, and when compared to designs obtained from a randomly selected training set, the recursive procedure produced improved designs. This is particularly noteworthy since only a fraction of the training data was used, and thus the cost in computing time and effort was significantly reduced. Given that the recursive procedure was able to isolate the five various optimal configurations in the first few iterations, IOP sets of less than 27 designs may have provided adequate results. This may be an indication that the necessary IOP size is a function of the design space nature (both in terms of the dimension and the inherent complexity of the design space).

The recursive learning procedure has shown its potential for use with design space representation. The impetus for developing a recursive procedure was to allow for the reduction of training data since the generation of training information was typically the most time consuming portion of the neural network utilization.

Table 17. Four Spar Recursive Configuration Design, Tip Displacement Based Designs

DESIGN OBJECTIVE	NN OPTIMAL SPAR LOCATIONS (% Chord)	W (lb)	δ (in.)	ω1 (Hz)	ω2 (Hz)	ω3 (Hz)
δ_{TMIN}	1) 16.4, 35.5, 60.0, 75.0	314.9	**7.83**	6.53	20.77	30.93
	6) 15.0, 38.6, 60.0, 75.0	317.5	**7.93**	6.46	20.84	30.85
	11) 22.4, 35.0, 60.0, 75.0	304.5	**7.83**	6.75	20.03	31.15
	16) 15.0, 35.0, 60.0, 75.0	317.4	**7.83**	6.48	20.96	30.84
	21) 16.8, 35.0, 60.0, 75.0	314.3	**7.81**	6.55	20.72	30.94
27-Random	20.0, 35.0, 59.2, 75.0	307.9	**7.86**	6.64	20.13	31.18
100-Random	21.2, 37.5, 56.8, 75.0	306.3	**7.90**	6.68	19.89	31.30

Table 18. Four Spar Recursive Configuration Design, First Natural Frequency Based Designs

DESIGN OBJECTIVE	NN OPTIMAL SPAR LOCATIONS (% Chord)	W (lb)	δ (in.)	ω1 (Hz)	ω2 (Hz)	ω3 (Hz)
ω_{1MIN}	3) 15.0, 35.0, 60.0. 75.0	317.7	7.81	**6.49**	20.94	30.83
	8) 15.0, 45.0, 53.1, 75.0	320.5	8.31	**6.32**	21.09	30.72
	13) 15.0, 45.0, 59.0, 75.0	319.9	8.31	**6.33**	21.49	30.51
	18) 15.0, 45.0, 55.9, 75.0	320.0	8.31	**6.32**	21.29	30.59
	23) 15.0, 45.0, 59.9, 75.0	319.8	8.31	**6.33**	21.55	30.49
27-Random	15.0, 40.0, 50.0, 75.0	312.3	8.00	**6.51**	20.31	31.21
100-Random	15.0, 40.6, 50.1, 75.0	319.1	8.08	**6.39**	20.40	31.17

Table 19. Four Spar Recursive Configuration Design, Second Natural Frequency Based Designs

DESIGN OBJECTIVE	NN OPTIMAL SPAR LOCATIONS (% Chord)	W (lb)	d (in.)	ω_1 (Hz)	ω_2 (Hz)	ω_3 (Hz)
ω_2MIN	4) 30.0,45.0, 50.0, 65.0	270.6	8.21	7.43	**14.57**	32.54
	9) 30.0, 41.1, 40.0, 65.0	270.3	8.12	7.46	**14.35**	32.76
	14) 30.0, 40.9, 50.0, 65.0	270.4	8.12	7.46	**14.38**	32.75
	19) 30.0, 41.2, 50.0, 65.0	270.4	8.12	7.46	**14.36**	32.75
	24) 30.0, 40.6, 50.0, 65.0	270.2	8.12	7.45	**14.34**	32.77
27-Random	20.4, 37.5, 51.9, 87.7	285.9	7.99	6.99	**17.11**	32.54
100-Random	20.2, 38.9, 50.0, 65.0	287.2	8.01	6.99	**17.01**	32.68

7.3 Material Selection - Discrete Design Variables

The next three design studies illustrate the application of simulated annealing and neural networks to discrete design variable problems. Each design study was modeled using discrete design variables in order to perform material selection for a given design configuration. The first problem is a ten bar truss material system design, in which four materials are considered for each of the ten rods. A least weight material system was the objective. The second problem is the ACOSS II space truss design, in which four materials were considered for each of the 113 rods. A least weight material system was again desired. The last, rather brief example, was an attempt to use this approach to select laminate orientations for the wing skins for a simple wing-box structure. It should be noted that the simulated annealing procedure was often able to determine designs that were a significant improvement over the "best" designs present in the neural network training set. No recursive learning was employed for these studies.

7.3.1 Ten Bar Truss

It was desired to isolate the material systems that provided least weight truss designs under the various constraints imposed. Four different network combinations were considered, composed of 20, 40, 80, and 120 neurons in the hidden layers. Four different sets of training data were also employed, composed of 20, 50, 100, and 200 designs (200 designs represent only 0.02% of all possible designs). Comparisons were made between the simulated annealing predictions for least weight material combinations for these various network geometries and IOP combinations against a set of exhaustive searches performed using these same networks. This comparison was made possible because of the neural network efficiency, which allowed the investigation of slightly over 1 million (actually, 4^{10}) material combinations.

Table 7 shows the four materials employed for the design of the ten bar truss. Two different grades of aluminum were considered, as well as steel and titanium. An exhaustive search was performed using the four different neural networks. This involved propagating all 4^{10} possible material systems through the networks and extracting the least weight material systems predicted by the networks. Figure 59 presents the four least weight material combinations for the exhaustive search performed with the 40-20-1 networks. The legend of this figure includes a set of four numbers in parenthesis. These numbers represent the number of times that the simulated annealing procedure identified the global minima that were obtained from the exhaustive searches. Since there were ten simulated annealing executions performed for each network/IOP combination, this number indicates the relative effectiveness of the simulated annealing control mechanism and generation mechanism, in isolating the global minimum. This number may also give a relative indication of the quality of the design space mapping, since the simulated annealing procedure (generation and control mechanisms) remains constant for all the networks. The other designs obtained from the ten simulated annealing optimizations were local minima, and were in most cases one of the five least weight designs obtained from the exhaustive search.

Regarding the material selections shown in Figure 59, it can be seen that there was good agreement between the materials selected for the four IOP cases. All ten rods appeared to have strong tendencies towards one material over the other three. Rod 1 clearly has an affinity with material 2, rod 2 for material 2, etc. It would be expected that as the IOP number increases the changes made to the material vector would improve the design characteristics (i.e., lower the design weight), since increasing IOP number had an associated improvement in the design space representation's quality. As will be shown shortly, this was indeed the case. Only rods 2, 3, and 9 did not show some change in material as the IOP number increased.

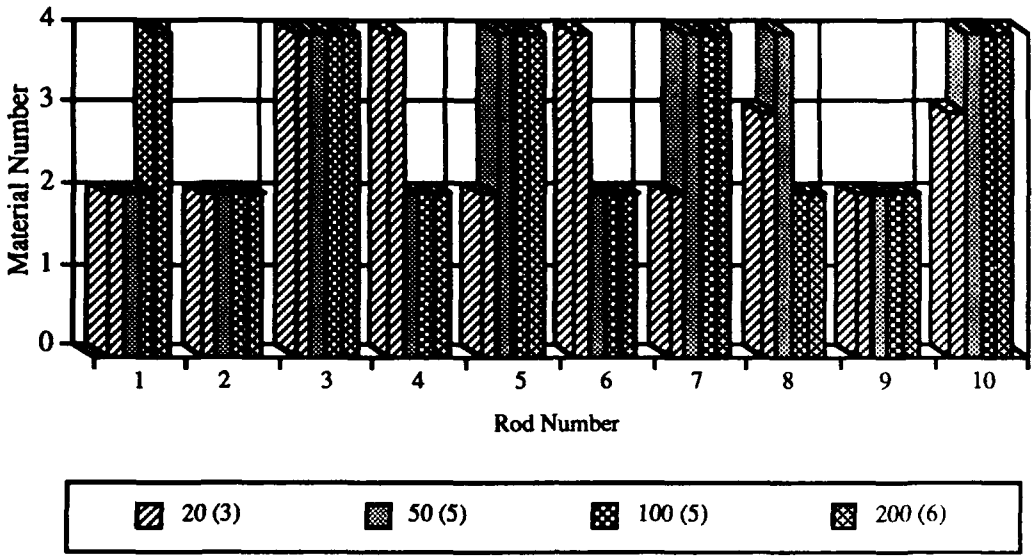


Figure 59. Ten Bar Truss Minimum Weight Material Selection Obtained from an Exhaustive Search (40-20-1 Network)

It was seen that as the IOP number increased, the simulated annealing procedure was better able to isolate the global minimum. For the 20 IOP case, the global minimum was isolated only three times, while for the 50 and 100 IOP cases the global minima were isolated five times. For the 200 IOP case, the minimum was isolated six times. Since the simulated annealing procedure was unchanged among these four cases, it appeared that the ability of the simulated annealing procedure to isolate the global minimum improved when applied using the neural networks trained with increased training set size. This is possibly due to the "improvement" of the design space representation. It was noted that when the simulated annealing procedure was not able to isolate the global minimum, that one of the five least weight material combinations (as determined from the exhaustive search) was usually identified. Thus, the simulated annealing procedure, while not guaranteeing that the global minimum will be identified, was able to identify at least one of the five lowest weight designs.

Figure 60 presents the weights obtained for the trusses designed using the 40-20-1 networks. The network predicted weight values were compared to the actual weight values obtained from ASTROS designs employing the same material combinations. There was a distinct reduction in weight as the IOP number increased, as well as an improvement in the agreement between the network predicted weight value and the actual ASTROS weight value. Thus, as the IOP number increased the design space "quality" appeared to improve and the ability of the simulated annealing procedure to extract improved designs also increased.

Figure 61 describes the four least weight material combinations obtained from the exhaustive searches performed on the 40-40-1 networks. Definite material assignment trends can be noted, although every rod has at least one material difference between the four IOP cases. There was some consistency in the number of times that the simulated annealing procedure was able to isolate the minimum weight material combinations. The 20, 50, and 200 IOP cases all isolated their respective minimums seven out of ten times, while the 100 IOP case isolated the global minimum only three times. It is uncertain why this occurred for the 100 IOP case. It was expected, based on the results from the 40-20-1 networks, that as IOP number increased that the prediction of the global minimum weight material system would improve. The "quality" of the design space representation should improve as the IOP number was increased.

Figure 62 shows a comparison between the neural network predicted weights for the global minimum material combinations shown in Figure 61 and the actual ASTROS weight values for these material systems. As was seen previously for the 40-20-1 network, there was a distinct reduction in weight as the IOP number increased. This decrease in weight was accompanied by an increase in accuracy for the neural network predicted weight values.

The four least weight material combinations obtained from the exhaustive search using the 40-80-1 networks are shown in Figure 63. Strong trends in material selection were again noted, with complete agreement over all IOP numbers occurring for rods 3, 5, 7, and 8. The number of times that the simulated annealing procedure isolates the global minima varied among the four IOP sets, with the lowest correspondence occurring for the 200 IOP case (with only three matches). The number of times that the global minimum was isolated showed trends similar to those of the 40-40-1 networks. The 200 IOP case for the 40-40-1 network also had the lowest occurrence of isolated global minima.

A comparison between the neural network-predicted weight values for the four minimum weight material combinations and the actual weight values is provided in Figure 64. The trend in weight reduction that was noted for the previous network geometries was less apparent, although the 200 IOP case was clearly the least weight design. The improved agreement that was noted between the network and actual weight values was also less apparent and the 200 IOP case actually showed a relatively poor agreement between weight values.

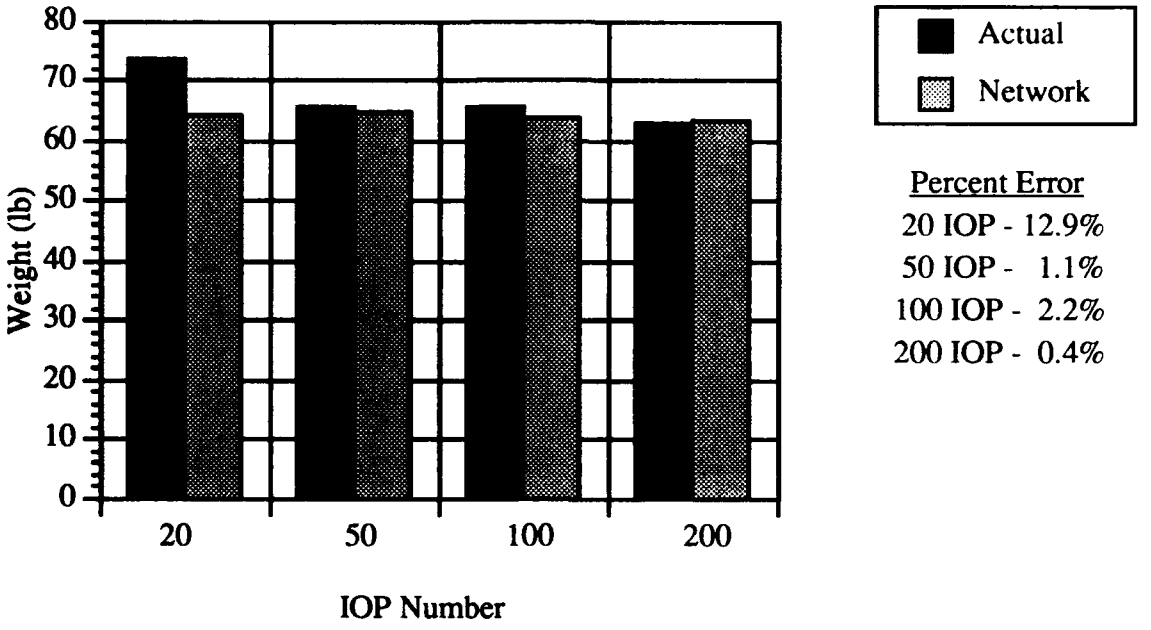


Figure 60. Ten Bar Truss Minimum Weight Material Selection Comparison of Actual Structure Weights to Neural Network Predictions (40-20-1 Network)

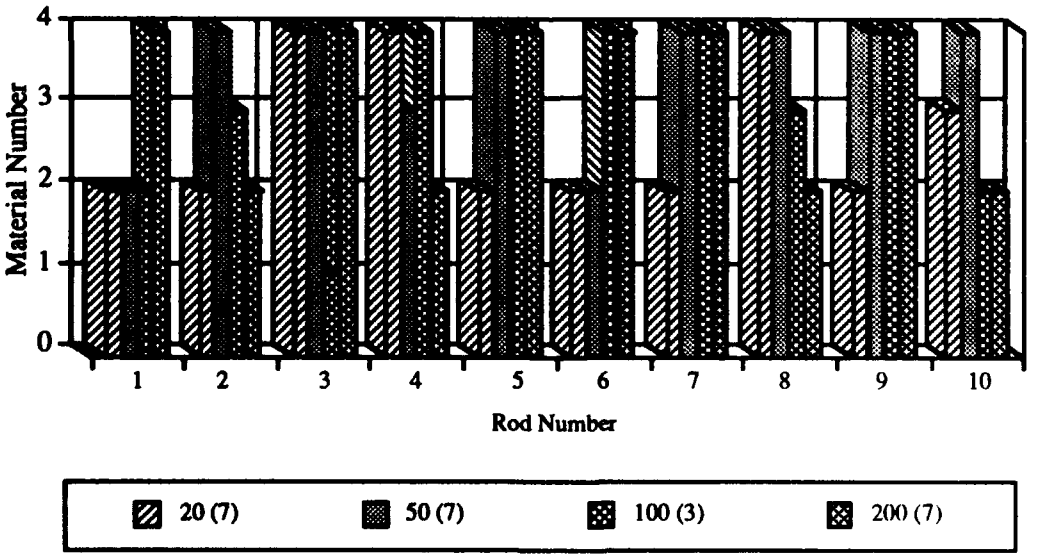


Figure 61. Ten Bar Truss Minimum Weight Material Selection Obtained from an Exhaustive Search (40-40-1 Network)

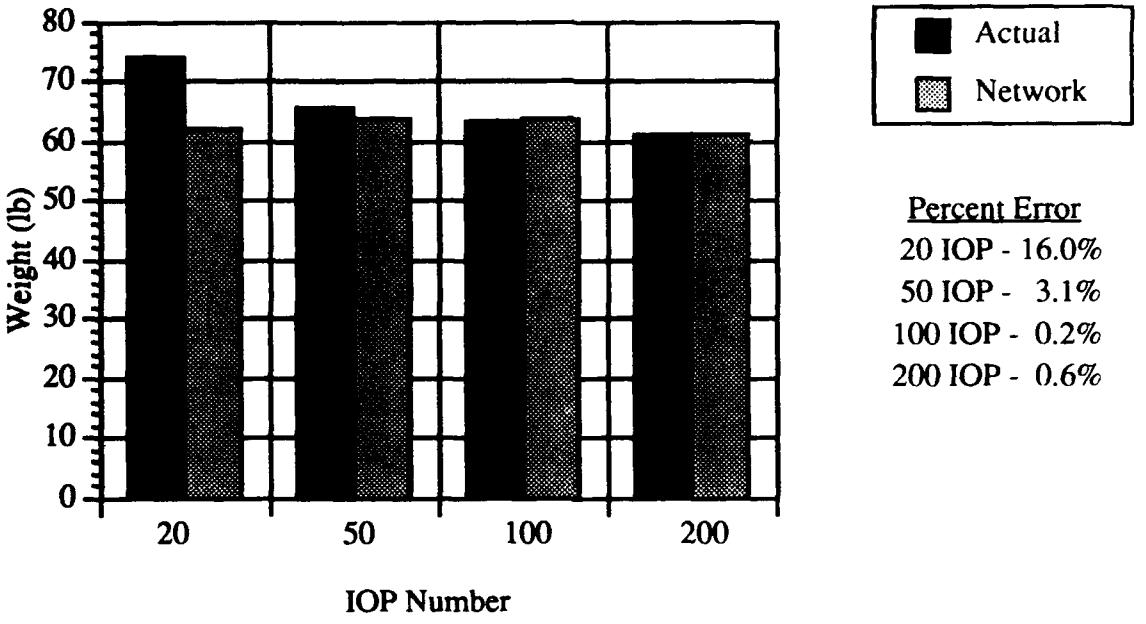


Figure 62. Ten Bar Truss Minimum Weight Material Selection Comparison of Actual Structure Weights to Neural Network Predictions (40-40-1 Network)

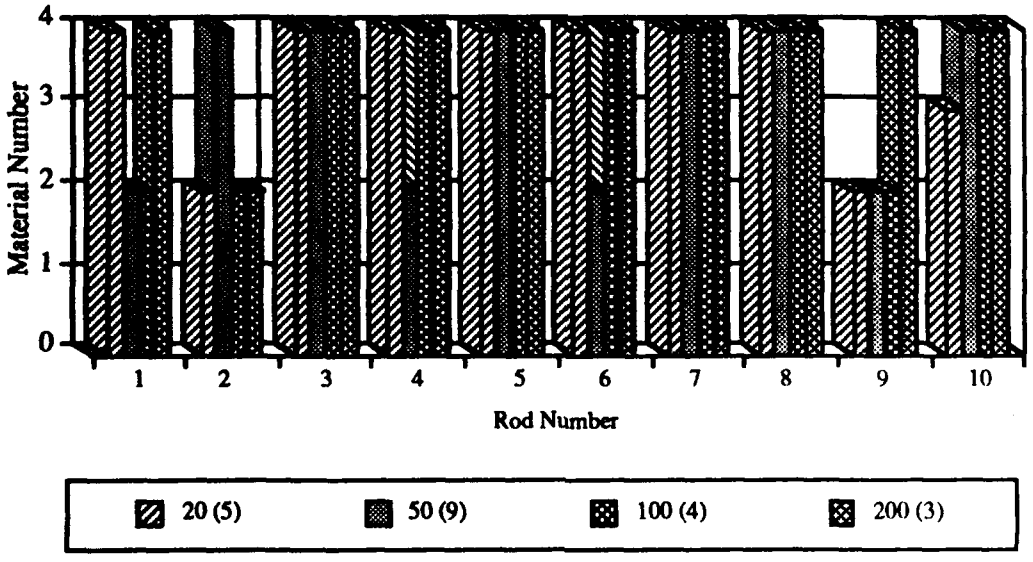


Figure 63. Ten Bar Truss Minimum Weight Material Selection Obtained from an Exhaustive Search (40-80-1 Network)

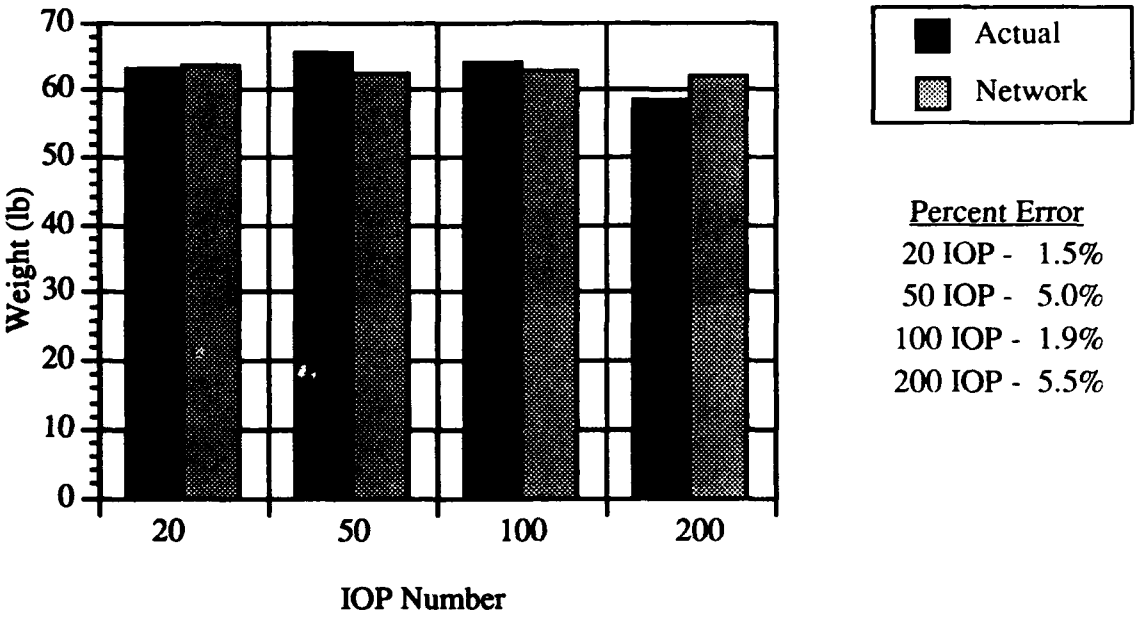


Figure 64. Ten Bar Truss Minimum Weight Material Selection Comparison of Actual Structure Weights to Neural Network Predictions (40-80-1 Network)

The results for the largest hidden layer size considered are shown in Figure 65. The four least weight material combinations obtained from the exhaustive search performed with the 40-120-1 networks are illustrated. Again strong material trends were noted, with rods 5 and 7 having consistent values for their materials. The number of times that the simulated annealing procedure was able to isolate the global minimum varied. The highest number of matches occurred for the 100 IOP case, while the lowest occurred for the 200 IOP case. As was seen for both the 40-80-1 and 40-40-1 networks, the 200 IOP case again provides the lowest number of isolated global minima. This is not necessarily an argument against higher IOP sets, though, since in every case the 200 IOP material systems do provide the least weight truss designs.

Figure 66 shows a comparison between the neural network predicted weight values for the four least weight material combinations and the actual design weights for these combinations. The trend towards a reduction in weight as IOP number increased was again noted, although there seemed to be a decrease in the accuracy of the neural network predicted weight values as the IOP number increases. This was counter to what would be expected.

Figure 67 shows the percent error trends for the actual design weights and the neural network predictions. Both the 40-20-1 and 40-40-1 networks showed a large error for the 20 IOP case. The 40-80-1 and 40-120-1 networks showed errors less than 1% for this case. This was surprising since the amount of training data was a very small subset of the 4^{10} possible material combinations. All the other network and IOP combinations displayed what might be termed a moderate error (in the range of 3-5%). It appeared that by increasing the number of neurons in the hidden layer that the ability of the network to accurately represent the design space improved since the average error values decreased with an increasing number of hidden layer neurons. The average error values for the four networks showed a decrease as the number of neurons increased, with the exception of the 40-40-1 neuron case which did show a slight error increase over the 40-20-1 average error value. The relatively complex interaction between training set size, neural network size and design space complexity is demonstrated by this result. The determination of the most appropriate network geometry and training set is a design problem in its own right!

Some important abilities of the neural networks were identified with this example. First, the neural networks were able to accurately represent the design space characteristics so that improved designs could be extracted. The typical reduction in weight obtained from the simulated annealing material selection process was better than 12% (as compared to the least weight truss design present in the training data). Second, the weight values for the truss predicted from the neural network were also quite accurate. Both of these accomplishments were made even more significant by the small subset of information used to train the neural network (only 200 out of a possible 4^{10} combinations), and the highly discrete, discontinuous nature of the design space.

7.3.2 ACOSS II

This example was meant to represent a realistic material system design problem, both in terms of complexity and computational requirements. The ACOSS II space truss was modeled using 113 axial force rod elements. The design had natural frequency constraints only. These were an upper bound constraint on frequency of 2.0 and 3.0 Hz respectively for the first and second modes. The design problem under consideration was again material assignment. The combinatorial size of all potential material systems is 4^{113} possibilities.

To investigate the influence of neural network geometry and IOP size on the simulated annealing approach three different networks were employed with four different levels of training data. Each of these networks had 452 inputs for the material vector and a single output corresponding to the design weight. The networks had 10, 40, and 100

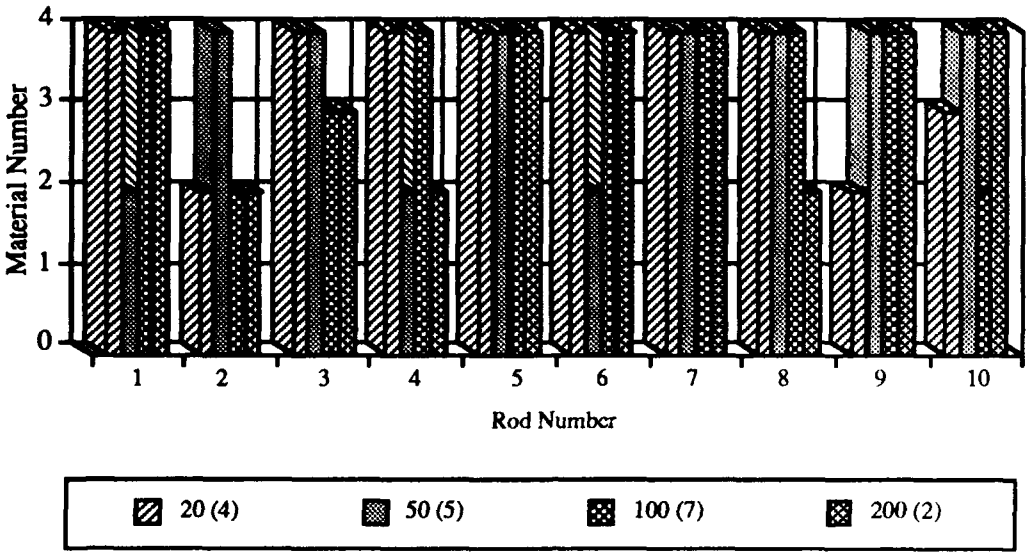


Figure 65. Ten Bar Truss Minimum Weight Material Selection Obtained from an Exhaustive Search (40-120-1 Network)

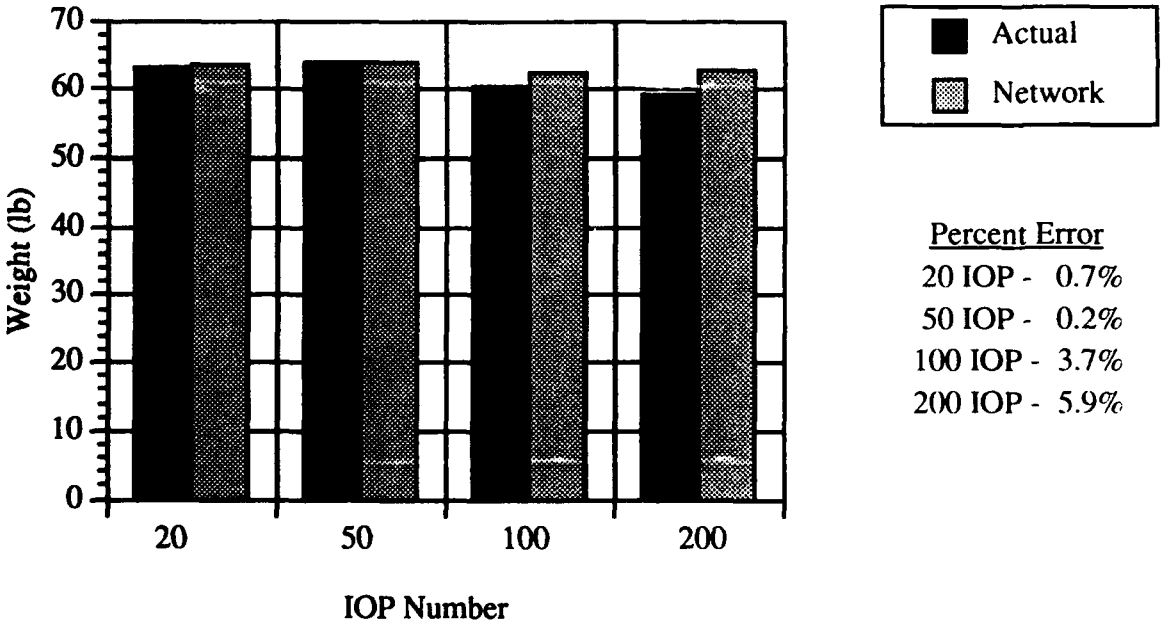


Figure 66. Ten Bar Truss Minimum Weight Material Selection Comparison of Actual Structure Weights to Neural Network Predictions (40-120-1 Network)

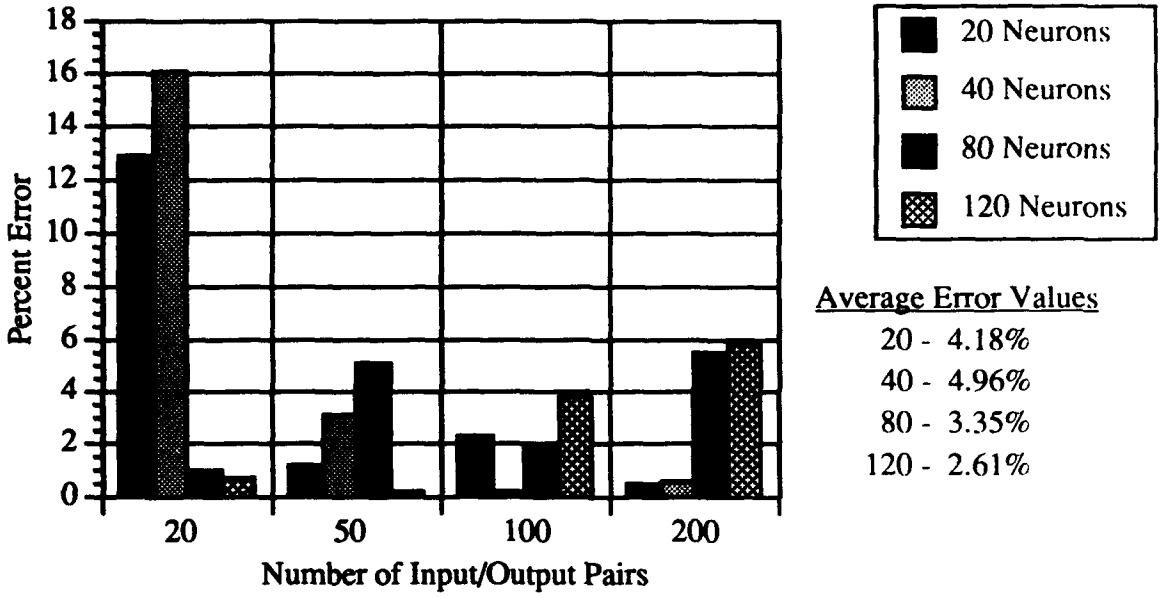


Figure 67. Ten Bar Truss Minimum Weight Material Selection Comparison of Percent Errors Between Network Predictions and Actual FSD Values

neurons in the single hidden layer. Training sets composed of 50, 100, 300, and 500 designs were used. Material systems were selected at random and then optimized using ASTROS. The networks were trained to a maximum error value of E=0.05 for all IOP sets. Values of E less than 0.05 were found to be difficult to obtain (a low back-propagation convergence rate was observed for cases where $E < 0.05$), particularly for the 300 and 500 IOP training sets. This termination error value did not seem to adversely effect the final network-predicted designs.

Because of the combinatorial size of the material design problem, an exhaustive analysis using the neural network could not be performed. It is estimated that this would take in excess of 10^{50} CPU years, based on approximately 10^{68} material combinations and an average neural network propagation time for a single material vector of less than ten milliseconds on a DECstation 3100. This limits the ability to discern the simulated annealing procedure's effectiveness in isolating the true global minimum weight design from the neural network's representation of the design space (as could be done for the ten bar truss).

It should be noted that the simulated annealing procedure had a fixed set of constants that defined the nature of the generation mechanism and the control parameter. Thus, the simulated annealing procedure was unchanged for each of the networks utilized. The nature of the neural network design space representation was what defined the success or failure of the existing simulated annealing procedure. The simulated annealing parameters that maximized the performance of the algorithm for the 500 IOP set were chosen.

Table 20 shows the characteristics of the four sets of training data that were employed for this study. For each of the four sets, the maximum and minimum weight designs were the same. The maximum weight design found for the 500 designs was 8761.4 lb and the minimum weight design was 6986.0 lb. This simplified the IOP scaling procedure by allowing all four IOP sets to have the same scaling factors. Materials for individual material vectors were selected at random, and it can be seen from Table 20 that the percentages of materials within the structures had a nearly uniform distribution. All materials averaged near 25% for the composition of the total material vectors for each IOP set. Individual materials comprised at least 12% of any given truss, and never exceed 39%. The average ASTROS design weight for each of the IOP sets was also comparable.

For each of the network and IOP combinations the simulated annealing procedure was executed five times, resulting in five different designs (a total of 60 designs for all network/IOP combinations). In each of these 60 cases, the simulated annealing procedure was run for 20,000 iterations and the least weight network prediction was retained. A typical simulated annealing iteration history is shown in Figure 68. The network used in this case was the 452-10-1 network with the 500 IOP training set. The scaled network output had an initial value just below 0.7 (a value obtained from an initial, randomly generated material distribution in the SA procedure). The SA procedure then rapidly converged to the low weight, final design. The inset for the figure describes the first 2000 iterations in more detail. For this example the lowest weight (scaled weight) design was obtained at iteration 13,515. This scaled weight value, when converted to an actual weight value, becomes 6825.5 lb. In contrast, the actual ASTROS weight for this material vector is 6077.9 lb. This large discrepancy between the network-predicted design weight and the actual ASTROS design weight was noted for all 60 executions of the SA algorithm.

The error between the network-predicted weights and the actual ASTROS design values can be attributed, at least partially, to the training data scaling range. The training data was scaled in the range of 0.1 to 0.9, which had proven reasonably successful in previous design studies. However, if the network could provide an output of "0" (the lowest possible neuron output), the corresponding design weight becomes only 6764.1 lb. The existing scaling, therefore, can never accurately extrapolate to weight values for the

Table 20. ACOSS II Material Design Study, IOP Characteristics

IOP and Material Number	Material Percentages	Standard Deviation -Materials	Minimum Material Count	Maximum Material Count	Average Weight(lb)	Standard Deviation Weight
1)	24.55	4.06	19	37		
2)	24.44	4.33	18	37	7871.4	319.67
50 3)	26.33	4.74	19	39		
4)	24.67	4.21	22	39		
1)	24.03	4.32	19	38		
2)	24.67	4.55	18	43	7876.8	299.0
100 3)	25.57	4.47	19	39		
4)	25.74	4.62	21	41		
1)	24.38	4.40	15	39		
2)	24.96	4.29	14	43	7864.2	287.3
300 3)	25.51	4.36	19	43		
4)	25.15	4.50	17	41		
1)	24.79	4.64	15	41		
2)	24.96	4.50	14	43	7887.2	300.4
500 3)	25.14	4.41	19	43		
4)	25.11	4.51	15	41		

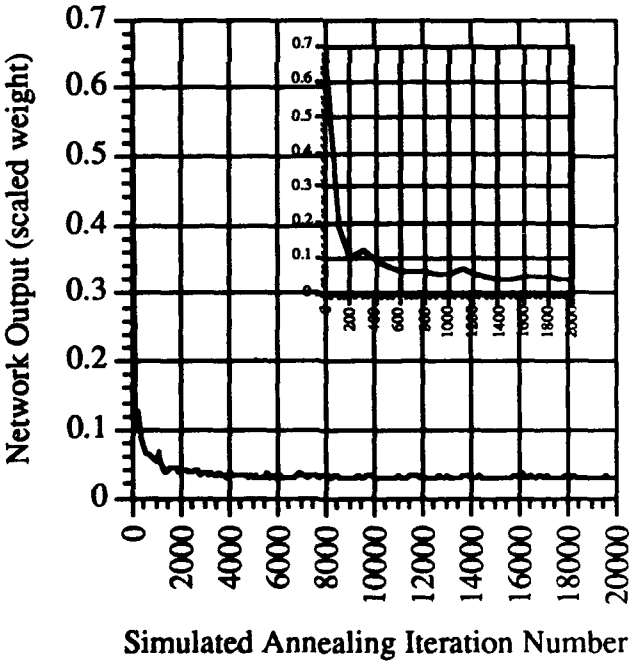


Figure 68. Typical Simulated Annealing Iteration History, ACOSS II Design Study

material systems that are significantly less than the lowest weight in the training set with the current scaling scheme.

Fortunately the trends in the design space were effectively mapped. Scaled training data in the range of 0.3 to 0.7 was also considered, and a better agreement between the network-predicted weight and actual ASTROS weight for several designs was noted (in the best case, the network predicted weight was 6491.2 lb versus a corresponding ASTROS design weight for the material system of 6130.5 lb). Scaling in the range of 0.3 - 0.7 gave designs that were heavier in weight (6130.5 lb also being the lowest actual ASTROS design weight isolated) than those obtained from the original scaling factors. The original scaling factors were retained due to this apparent better extrapolation ability even though the actual weight prediction error from the neural network was rather large.

The ASTROS weights and material percentages for the ACOSS II truss designs obtained from the 452-10-1 networks are shown in Table 21. Some distinct trends in the material vectors could be seen as the IOP number was increased. For the 50 IOP cases, it was seen that the percentages of the four materials in the five designs were quite similar, showing only a slight variation (less than $\pm$ 2% around the mean value). As the IOP number increased, there was a clear increase in the percentage of material 3 that composed the truss. There was also a dramatic decrease in weight for the resulting designs. Figure 69 shows the average material percentages for the 452-10-1 network designs as a function of the IOP number. The percentage of material 3 showed a slight decline between the 50 and 100 IOP cases, but then increased to better than twice its original value for the 200 IOP case. Material 3 had the highest specific stiffness, and due to the frequency constraints that drive the ACOSS design, a truss composed solely of this material provided the lowest weight design that has been achieved (6035.1 lb) for any of the designs analyzed.

Table 22 shows the material percentages and associated weights for the designs obtained from the 452-40-1 networks. The same trends that were noted for the previous network geometry were again seen, most notably the increase in the percentage of material 3 as IOP number increased. This increase resulted in an associated reduction in the design weight. The average material percentages obtained from the 452-40-1 network designs is illustrated in Figure 70. There was a clear dominance of material 3 for the designs as the IOP number increased, as expected from the results obtained from the 452-10-1 networks.

The results for the 452-100-1 networks are presented in Figure 71 and the associated tabular results are in Table 23. As had been noted for the previous two networks, the trend towards an increased percentage of material 3 in the resulting designs as the IOP number increased holds for this case. There was also a dramatic reduction in weight as the IOP number increased (and the proportion of material 3 became dominant).

The average percentages of the four materials as a function of the IOP number are presented in Figure 71. Again, the dominance of material 3 becomes clear as the IOP number increases. For each of the three network geometries, there was a dramatic increase in the proportion of material 3 assigned to the truss designs as the IOP number increased. This increase in the use of material 3 has an associated reduction in the design weight (averaging over 800 lb.). Clearly, the neural network representation of the design space is such that the weight trends associated with the material systems are being defined appropriately. The design space definition is clearly shown to be a function of the IOP number.

Figure 72 shows the average design weights obtained for the three networks at various IOP levels. All three networks gave very similar results in terms of the design weights obtained for various IOP levels. The most significant trend was the reduction in design weight as the IOP number increased. It was not expected that further significant reductions in weight could be obtained by further increasing the IOP level, since the

Table 21. Actual Weights, Material Percentages and Active Gage Constraints for the ACOSS II Material Design Study (452-10-1 Network)

Network Hidden-IOP	Weight (lb)	Material 1 Percent	Material 2 Percent	Material 3 Percent	Material 4 Percent	Min. Gauge Elements
	6816.4	21.2	30.1	31.0	17.7	36 31.9%
	6917.0	23.0	30.9	28.3	17.7	25 22.1%
	6808.7	22.1	31.9	27.4	18.6	27 23.9%
10-50	6887.0	23.9	29.2	28.3	18.6	33 29.2%
	6922.4	23.0	31.9	28.3	16.8	28 24.8%
Average	**6870.3**	**22.6**	**30.8**	**28.7**	**17.9**	**30 21.6%**
	6654.5	21.2	31.9	25.7	21.2	30 26.5%
	6627.6	19.5	30.1	25.7	24.8	30 26.5%
	6730.5	22.1	28.3	24.8	24.8	30 26.5%
10-100	6666.2	20.4	31.9	24.8	23.0	33 29.2%
	6645.7	18.6	32.7	25.7	23.0	30 26.5%
Average	**6659.5**	**20.4**	**31.0**	**25.3**	**23.4**	**31 27.0%**
	6278.3	17.7	17.7	44.2	20.4	30 26.5%
	6262.6	16.8	18.6	47.8	16.8	27 23.9%
	6291.4	17.7	17.7	43.4	21.2	24 21.2%
10-300	6297.7	16.8	17.7	46.0	19.5	29 25.7%
	6294.8	16.8	17.7	42.5	23.0	27 23.9%
Average	**6284.9**	**17.2**	**17.9**	**44.8**	**20.2**	**27 24.2%**
	6098.7	16.8	13.3	54.0	15.9	31 27.0%
	6084.7	14.2	15.9	56.6	13.3	32 28.3%
	6089.6	14.2	14.2	57.5	14.2	31 27.0%
10-500	6078.1	17.7	14.2	56.6	11.5	31 27.0%
	6077.9	14.2	15.0	57.5	13.3	32 28.3%
Average	**6085.8**	**15.4**	**14.5**	**56.4**	**13.6**	**31 27.5%**

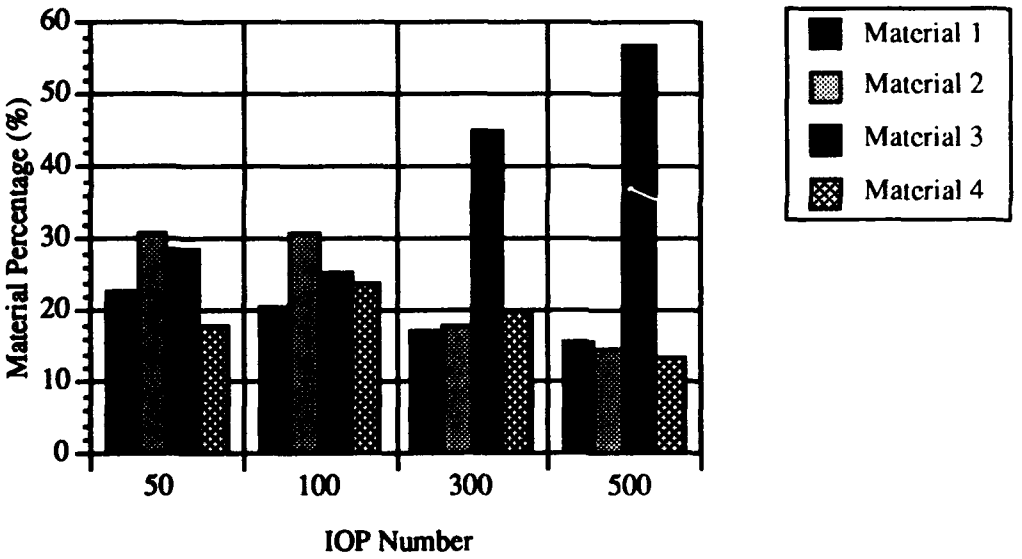


Figure 69. Distribution of Material Types, ACOSS II Design Study (452-10-1 Network)

Table 22. Actual Weights, Material Percentages and Active Gage Constraints for the ACOSS II Material Design Study (452-40-1 Network)

Network Hidden-IOP	Weight (lb)	Material 1 Percent	Material 2 Percent	Material 3 Percent	Material 4 Percent	Min. Gauge Elements
40-50	6943.0	20.4	27.4	27.4	24.8	29 25.7%
	6994.4	23.0	25.7	28.3	23.0	26 23.0%
	6846.9	22.1	28.3	25.7	23.9	29 25.7%
	6861.6	20.4	31.9	26.5	21.2	29 25.7%
	6803.6	21.2	28.3	27.4	23.0	34 30.1%
Average	**6889.9**	**21.4**	**28.3**	**27.1**	**23.2**	**29 26.0%**
40-100	6690.6	15.0	29.2	33.6	22.1	29 25.7%
	6746.3	16.8	29.2	30.1	23.9	30 26.5%
	6830.6	18.6	27.4	31.9	22.1	32 28.3%
	6726.7	18.6	24.8	32.7	23.9	28 24.8%
	6713.5	16.8	24.8	33.6	24.8	31 27.4%
Average	**6741.5**	**17.2**	**27.1**	**32.4**	**23.4**	**30 26.5%**
40-300	6230.6	15.9	16.8	46.0	21.2	29 25.7%
	6253.8	13.3	22.1	45.1	19.5	27 23.9%
	6239.5	15.9	15.0	45.1	23.9	27 23.9%
	6236.9	14.2	17.7	49.6	18.6	38 33.6%
	6324.7	16.8	18.6	41.6	23.0	26 23.0%
Average	**6257.1**	**15.2**	**18.0**	**45.5**	**21.2**	**29 26.0%**
40-500	6118.4	13.3	15.9	54.9	15.9	32 28.3%
	6110.8	20.4	14.2	56.6	8.8	28 24.8%
	6104.3	15.0	15.9	56.6	12.4	36 31.9%
	6125.4	17.7	16.8	53.1	12.4	32 28.3%
	6096.9	14.2	12.4	60.2	13.3	32 28.3%
Average	**6111.2**	**16.1**	**15.0**	**56.3**	**12.6**	**32 28.3%**

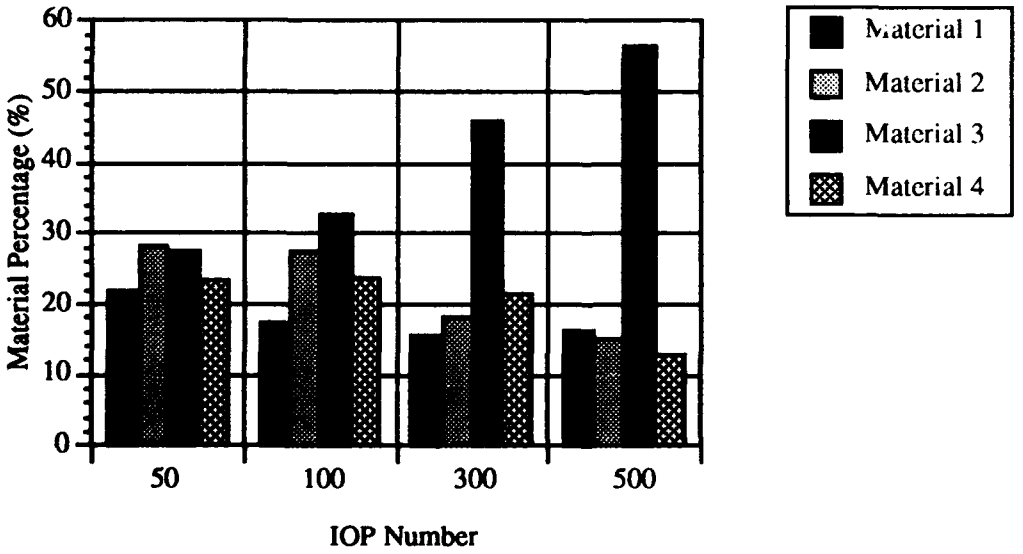


Figure 70. Distribution of Material Types, ACOSS II Design Study (452-40-1 Network)

Table 23. Actual Weights, Material Percentages and Active Gage Constraints for the ACOSS II Material Design Study (452-100-1 Network)

Network Hidden-IOP	Weight (lb)	Material 1 Percent	Material 2 Percent	Material 3 Percent	Material 4 Percent	Min. Gauge Elements
100-50	7166.4	20.4	31.9	23.0	24.8	30 26.5%
	7066.9	20.4	30.1	30.1	19.5	31 27.4%
	6985.2	21.2	32.7	29.2	16.8	36 31.9%
	7011.0	18.6	34.5	24.8	22.1	31 27.4%
	6989.1	21.2	27.4	26.5	24.8	29 25.7%
Average	**7043.7**	**20.4**	**31.3**	**26.7**	**21.6**	**31 27.8%**
100-100	6802.9	20.4	29.2	31.0	19.5	27 23.9%
	6833.8	18.6	32.7	29.2	19.5	32 28.3%
	6675.8	20.4	25.7	31.0	23.0	27 23.9%
	6684.5	19.5	31.9	26.5	22.1	27 23.9%
	6762.3	17.7	23.9	35.4	23.0	29 25.7%
Average	**6751.9**	**19.3**	**28.7**	**30.6**	**21.4**	**28 25.1%**
100-300	6251.0	18.6	15.9	44.2	21.2	32 28.3%
	6230.2	15.9	17.7	47.8	18.6	26 23.0%
	6319.8	15.9	20.4	42.5	21.2	25 22.1%
	6251.1	17.7	15.0	46.0	21.2	28 24.8%
	6273.6	16.8	18.6	45.1	19.5	26 23.0%
Average	**6265.1**	**17.0**	**17.5**	**45.1**	**20.3**	**27 24.2%**
100-500	6104.1	12.4	14.2	55.8	17.7	29 25.7%
	6110.9	12.4	20.4	54.9	12.4	34 30.1%
	6099.3	14.2	17.7	58.4	9.7	33 29.2%
	6117.1	15.0	18.6	52.2	14.2	31 27.4%
	6100.5	14.2	12.4	57.5	15.9	28 24.8%
Average	**6106.4**	**13.6**	**16.7**	**55.8**	**14.0**	**31 27.4%**

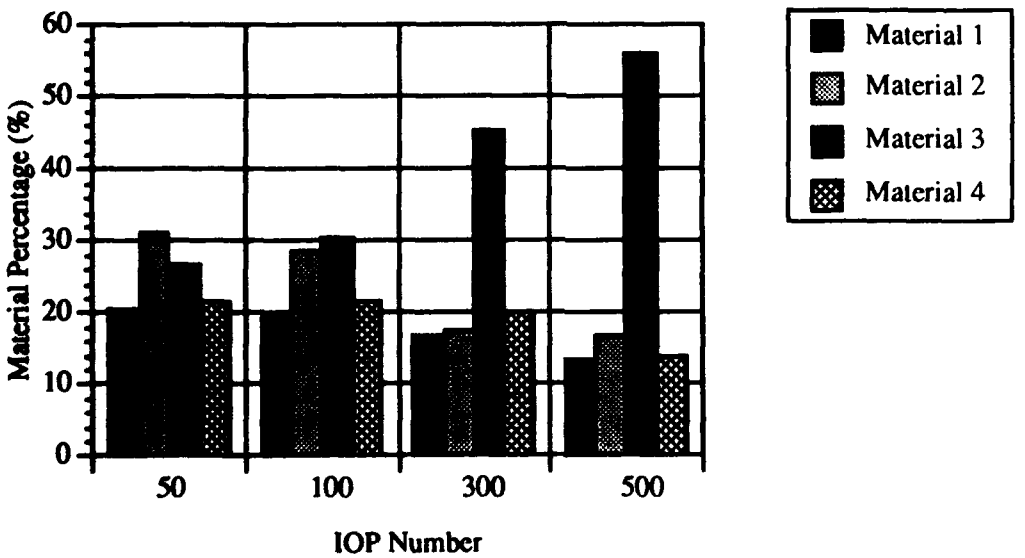


Figure 71. Material Percentages as a Function of IOP, ACOSS II Design Study (452-100-1 Network)

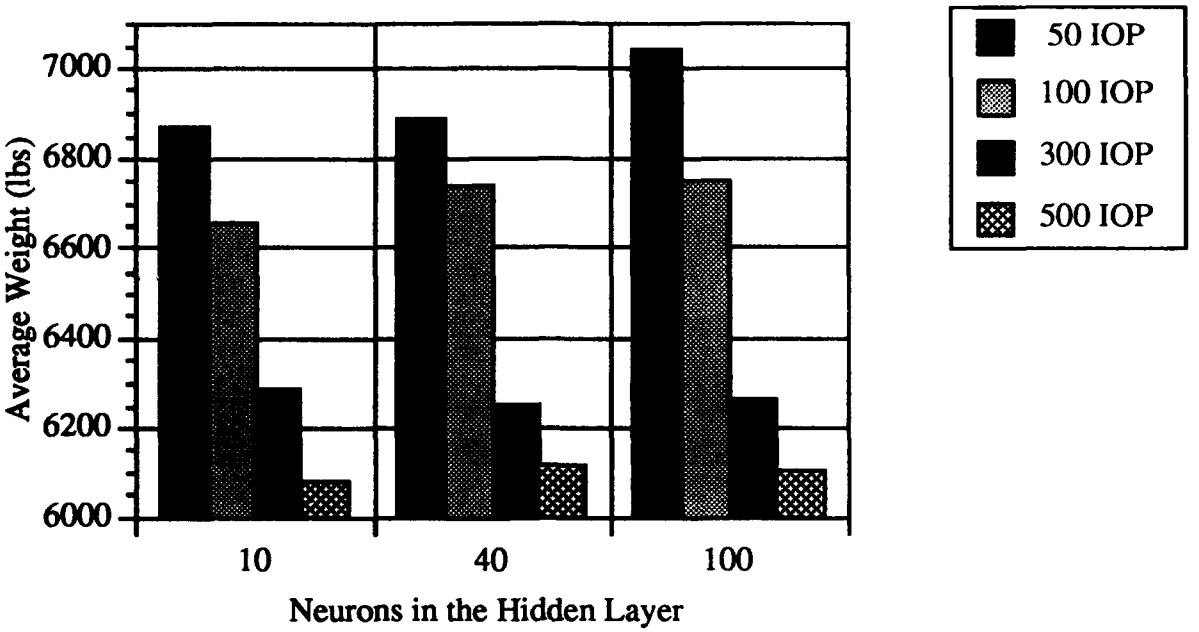


Figure 72. Average Actual Weights as Functions of IOP and Network Geometry, ACOSS II Design Study

expected least weight design (composed entirely of material 3) weighed only a fraction of a percent less than the least weight design obtained from simulated annealing.

It should be noted that all network predicted designs provided a weight reduction over the least weight design in the training set (6986.0 lb.). The average weight reduction was over 6% and the maximum weight reduction was over 12% (6077.9 lb).

The two least weight design material sets for the ACOSS II (as determined from simulated annealing) are shown in Table 24. The designs are very similar, with only 9 of the 113 rod elements being composed of different materials. These dissimilar rod materials are highlighted in bold in Table 24. Elements that are underlined are elements that have been designed to minimum gauge. Both designs share the same set of rods that have active minimum gauge constraints. Also, both of these designs were obtained from the 452-10-1 network trained with 500 IOP.

This design was governed by the two natural frequency constraints imposed. No stress constraints were used. It would thus be reasonable to expect that the material with the highest specific stiffness would provide a low weight design. This was the case. Material 3, graphite/epoxy type I, had the highest specific stiffness and a truss composed solely of this material had an ASTROS design weight of 6035.1 lb. This was lower in weight than either of the two designs shown in Table 24. The designs shown in Table 24 do have a large proportion of the elements composed of material 3, however. Design 1 had 65 rods composed of material 3, while design 2 had 64 rods that were composed of material 3. Given the combinatorial size of the design space it is likely that there are many material combinations that fall within a small percent difference of the lowest weight design observed thus far (composed entirely of material 3). The fact that the simulated annealing procedure has been able to identify designs that are within a small percent difference of this design simply further reinforces the potential of the neural network approach.

7.3.3 Multi-Spar Composite Wing-Box

This particular example was considered as a preliminary attempt to apply the concept of discrete design space modeling to composite structures. The neural network was trained to within 5% error and was then incorporated into the simulated annealing optimization procedure. The values for first natural frequencies in the training data ranged from 19.07 to 24.93 Hz.

The simulated annealing procedure was executed, and the five sets of lamina orientations predicted to provide lowest first natural frequencies were extracted. These lamina orientations were used for five additional ASTROS ICW designs, and the results are shown in Table 25. Of the five designs, only one did not provide an improvement over the training data (wing 3, with a first natural frequency of 19.16 Hz). The most significant improvement was with wing 5 where better than an 8% reduction in first natural frequency was achieved. The average reduction was better than 4%.

7.4 Design for Survivability

Two distinctly different approaches, and examples, were considered. The first involved the configurational design of a helicopter tail-boom in which six possible damage conditions were considered. A fully-stressed design procedure was used to achieve minimum weight designs while satisfying all applicable stress constraints. In the second example an undamaged baseline wing was designed under flutter and stress constraints for two aerodynamic loads. One hundred and fifty possible damage states were analyzed for flutter and natural frequency characteristics. These results were used to train a neural network. The neural network was then used to predict flutter occurrence and the natural frequencies for all possible damage conditions. A comparison between the neural network and 50 damaged wings was performed.

Table 24. Lowest Weight Material Configurations for the ACOSS II Material Design Study

Rod Numbers	Rod Materials for Design #1 6077.9 lb.	Rod Materials for Design #2 6078.1 lb.
1-10	2 1 3 3 1 3 4 1 3 3	2 1 3 3 1 3 1 3 3 3
11-20	3 4 3 2 3 2 1 3 3 2	1 4 3 2 3 3 1 3 3 2
21-30	3 3 3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3 3 3
31-40	3 3 2 4 3 3 3 1 1 4	3 3 1 4 3 3 3 1 1 4
41-50	3 4 1 3 3 3 3 3 3 3	3 4 1 3 3 3 3 3 3 3
51-60	2 3 3 3 3 3 3 3 1 2	2 3 3 3 3 3 3 3 1 2
61-70	3 4 1 3 3 3 3 1 4 4	3 4 1 3 3 3 3 1 4 4
71-80	2 3 2 4 3 3 3 3 3 1	2 3 2 2 3 3 3 3 3 1
81-90	4 4 3 3 1 2 3 3 3 2	4 4 3 1 1 2 3 3 3 2
91-100	3 4 3 1 3 3 4 4 3 2	3 1 3 1 4 3 4 4 3 2
101-110	3 1 1 1 2 3 2 3 3 2	3 1 1 1 2 3 2 3 3 2
111-113	4 2 2	4 2 2

Table 25. Resulting orientations for the ICW - SA designs

Lamina Orientations (degrees)	First Natural Frequency (HZ)
1. 65 0 -5 80	18.39
2. 75 0 -5 10	18.42
3. 75 40 5 20	19.16
4. 75 0 -5 -10	18.75
5. 75 40 0 -35	17.52

7.4.1 Helicopter Tail-Boom

As one might expect the analyses used to develop the training data produced a significant amount of information. At the preliminary design stage one could be overwhelmed by this information and must provide an efficient means for "storing" and retrieving it in a useful fashion. The identification of important design "trends" is critical. As an example, Figure 73 shows the variation in the third natural frequency for the damage tolerant designs as a function of section X_B location (other section locations remain fixed at X_A=56.5", X_C=133.5", X_D=146.5"). Due to the multi-dimensional nature of the design space, graphical presentation of the variation in system behavior with each design variable is more difficult and less useful. The values indicated as "NN" (neural network values) for the third natural frequency were obtained by simply determining the output from the trained neural net for a range of Section B locations. The data labeled as "FSD Values" are also for the third natural frequency but were determined for the same truss geometries using a complete FSD optimization for the fixed structural configuration. It should be noted that the training set used to develop the neural network did not contain any of these specific designs.

It is seen that the neural network was able to predict accurately the trends in frequency as a function of the design variables (in this case, to within 1% error). Because of the computational efficiency of the neural network analysis (better than two orders of magnitude faster than the FSD design in this example), a large amount of information could be extracted to determine trends for specified design variables or to determine those design variables which were dominant for certain structural characteristics.

The trained neural network was then employed as a subroutine to a math programming optimization procedure. Table 26 shows a comparison of several "optimal designs" predicted using the neural network. Four different cases were considered, including minimum weight and minimum natural frequencies for the first three natural frequencies. The system characteristics for the optima as determined by the neural net (subscript NN) are compared with the values corresponding to an FSD analysis of the "optimum" as selected by the neural network (subscript FSD). Recall that these are the "optimum" configurations as selected from families of "optimum" FSD designs. Table 27 presents a similar set of optimum designs as selected from the complete set of training data. Since the set of training data was relatively large, the optimum designs as selected from this set are rather close to those identified using the neural network. As one example consider the minimum weight configuration (W_{MIN}). The least weight configuration predicted using the neural network represented only a slight improvement over the least weight truss used in network training (77.48 lb versus 77.81 lb, respectively). All objective functions show a similar slight improvement. The agreement between the neural network prediction for all values correspond to well within 1% of the actual FSD values for a truss of the same geometry. Excellent agreement is present for all of the objective functions considered in Table 26. Note that the minimum first and second natural frequencies occur for the same truss geometry.

The neural network was able to accurately represent the design space of the helicopter tail-boom. Due to the nature of the design space and the size of the training set, significant improvements in the objective function(s) were not obtained in this particular example. The neural network may be used as an accurate and computationally efficient alternative to the FSD procedure to determine trends in all of the structural response characteristics to the configurational variables. Reducing the size of the training set or using the recursive training approach may result in more significant computational savings.

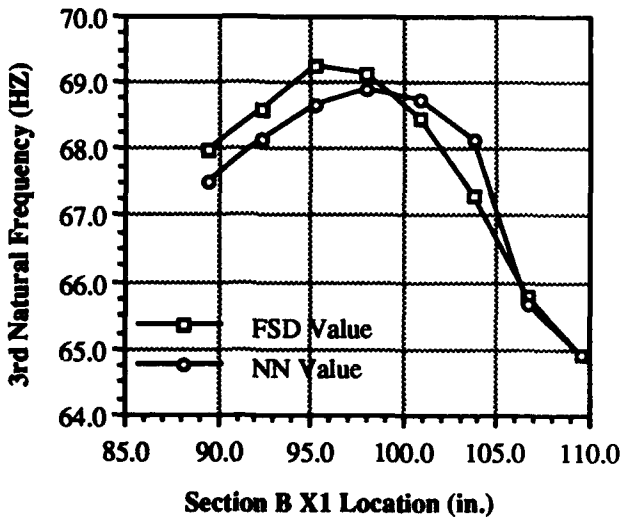


Figure 73. Variation in Third Natural Frequency with Section B Location, Helicopter Tail Boom Design for Survivability

Table 26. A comparison of the neural network predicted optimums and the actual FSD values for the helicopter tail-boom

DESIGN OBJECTIVE	NN OPTIMAL SECTION LOCATIONS (X_A, X_B, X_C, X_D in inches)	W_{NN} (lb)	W_{FSD} (lb)	ω_{1NN} (Hz)	ω_{1FSD} (Hz)	ω_{2NN} (Hz)	ω_{2FSD} (Hz)	ω_{3NN} (Hz)	ω_{3FSD} (Hz)
W_{MIN}	67.2, 97.3, 122.3, 148.7	77.66	77.48	21.23	21.25	26.06	26.14	74.17	74.13
ω_{1MIN}	56.5, 89.5, 133.5, 146.5	85.48	86.37	20.19	19.66	24.57	23.94	67.51	67.99
ω_{2MIN}	56.5, 89.5, 133.5, 146.5	85.48	86.37	20.19	19.66	24.57	23.94	67.51	67.99
ω_{3MIN}	56.5, 109.5, 133.5, 146.5	86.73	87.99	20.24	19.96	24.75	24.47	64.93	63.90

Table 27. Optimum designs as selected from training data

Training Data	Section Locations (X_A, X_B, X_C, X_D in inches)	W (lb)	ω_1 (Hz)	ω_2 (Hz)	ω_3 (Hz)
W_{MIN}	65.2,98.5,123.9,147.6	77.81	21.17	26.04	73.86
ω_{1MIN}	67.3,91.7,133.1,147.2	83.04	20.35	24.91	69.32
ω_{2MIN}	67.9,89.6,131.7,146.6	83.67	20.35	24.79	68.94
ω_{3MIN}	57.0,108.7,129.6,147.7	86.01	20.42	25.02	65.74

7.4.2 Five Spar Wing-Box

Of the 150 damaged models used for network training, only eight were damaged in such a manner that the flutter speed was reduced to below the design mach number of 0.85. In almost every case, flutter occurred in mode 2 at approximately M = 0.75. This therefore represented a very sparse set of training data and the use of this network for predicting the flutter characteristics for additional damage cases was not expected to be very accurate. Indeed this was the case. When considering the additional 50 damaged wings used for a comparison with the neural network model, only two exhibited flutter. In the first case, the neural network predicted that flutter would occur (although the flutter speed prediction was in error, 869 ft/s versus 650 ft/s for the ASTROS value and the network prediction, respectively). For the second flutter case, the neural network predicted that flutter would not occur. The neural network's ability to map the relationships between the damaged elements and the flutter characteristics for such a small set of training data proved to be inadequate. This was also indicative of the complexity of the flutter design space.

Figure 74 illustrates the prediction of natural frequencies obtained for a specific damage condition. This is the same model described previously that predicted an accurate occurrence of flutter. The agreement between the actual natural frequencies for this damage case and the neural network predictions is illustrated. Table 28 provides more detailed information on the accuracy of the neural network and its ability to predict the dynamic characteristics of the damaged wing-boxes. The data shown in the table represent a percent error and were obtained by a comparison with the 50 damaged wings not present in the neural network training data. Multiple neural network geometries were considered in order to begin to evaluate the effect of neural network geometry on the damage model accuracy. The neural network geometry represents the number of input nodes, number of nodes in the hidden layer(s), and the number of output nodes. The data labeled $\omega_{i\ RMS}$ for i= 1-3, are the root mean square errors between the neural network prediction and the actual natural frequency for the 50 damaged conditions not included in the training data. The data labeled $\omega_{i_{MAX}}$ are the largest errors for the complete set of 50 cases. The RMS_{TOT} and MAX_{TOT} represent summaries of the error for all cases and all three natural frequencies. For all the networks considered, the total RMS error (RMS_{TOT}) for the first three natural frequencies falls in the range of 4-7%. Maximum errors with the damage cases not included in the training set are as high as 23%. The 130-24-24-5 network provides the best mapping of the damage conditions. For this network, RMS_{TOT} = 4.62% and MAX_{TOT} = 12.18%. Additional training data may reduce this error but the current data indicated that the neural network represented the dynamic characteristics in a reasonable fashion for a variety of neural network geometries.

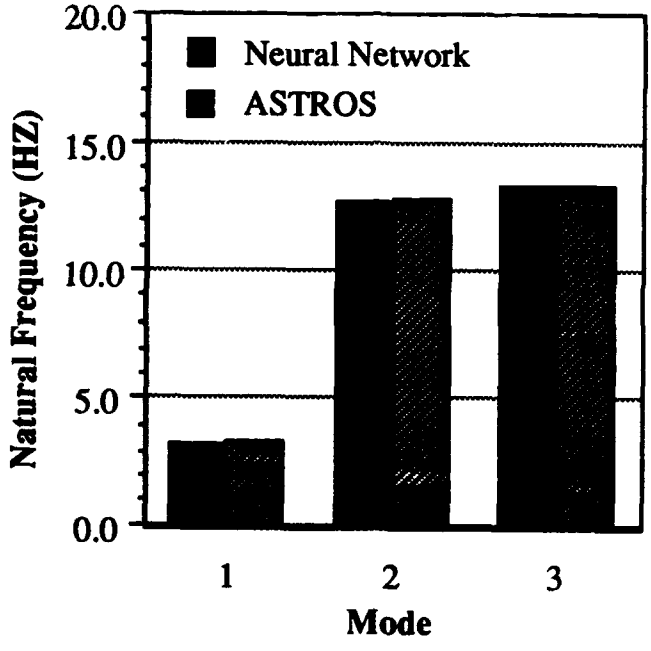


Figure 74. Comparison between Network Predictions and ASTROS, Damage Tolerance of a Multi-spar Wing Box

Table 28. Results for the wing-box damage study

NEURAL NETWORK GEOMETRY	ω_{1RMS}	ω_{1MAX}	ω_{2RMS}	ω_{2MAX}	ω_{3RMS}	ω_{3MAX}	RMS_{TOT}	MAX_{TOT}
130-12-5	8.47	22.61	8.05	23.31	3.27	11.28	7.01	23.31
130-16-5	6.10	14.58	5.18	10.92	3.17	10.34	4.97	14.58
130-24-5	6.19	14.12	5.39	19.42	3.20	10.42	5.09	19.42
130-36-5	5.64	15.23	5.25	14.21	3.00	9.54	4.77	15.23
130-48-5	5.81	13.63	5.23	14.79	3.16	8.69	4.86	14.79
130-12-12-5	8.98	22.84	5.11	15.99	3.14	9.79	6.23	22.84
130-16-8-5	7.09	22.66	4.91	10.85	3.14	10.66	5.30	22.66
130-18-6-5	5.71	13.79	4.61	12.08	3.33	8.09	4.65	13.79
130-24-24-5	4.97	12.18	4.14	10.24	3.18	10.11	4.16	12.18
130-36-12-5	5.61	13.15	4.76	12.98	3.17	10.32	4.62	13.15

8. SUMMARY OF RESULTS AND CONCLUSIONS

This report documents the development of methods and a series of design studies performed to evaluate the application of finite element-based analysis and numerical optimization methods to the preliminary design of aerospace structures. In each design study, an MDO procedure was used to develop design information. In most cases this information was detail on an "optimum" design of a fixed structural configuration and materials. This information was then used to train an artificial neural network to represent the design space for the particular problem. The neural network provided a quantitative representation of various measures of merit or performance indices as functions of specific design variables. In a number of cases, the neural network was then used to identify potential improved design concepts. The following summarizes a number of important result for each of these studies. These are followed by some general conclusions and recommendations.

8.1 Configurational Design with Continuous Design Variables

Configurational design involved the development of approaches that could lead to the determination internal arrangements of structural components. This in turn could provide improved design characteristics (i.e., better performance, lowered cost, lowered weight). The ability of the neural network to effectively represent the design information and extract improved designs was established.

8.1.1 Ten Bar Truss

A planar ten bar truss was chosen for its simplicity (in terms of the structural and design models). A test matrix of 24 different network geometries, IOP sets, and termination error combinations was investigated. This involved two different neural networks, six different sets of training data, and two different levels of training termination error. The two neural networks were distinguished by the number of neurons in the single hidden layers. The sets of training data ranged from 50 to 500 fully stressed designs. The two training termination criteria were considered. To make determinations on the design space representation ability of these various combinations, an additional set of 500 FSD truss designs was used to see how accurately the neural networks were able to represent this new design information. A number of trends were noted from these various cases:

a. The maximum and RMS error values for the two network geometries, determined by comparison with the 500 new designs, both showed a trend towards lowered error values as the number of training pairs was increased. This is as would be expected, since the design space became further refined as new design information was added to it.

b. When training termination error was reduced, an increase in both maximum and RMS error was noted. The network was more accurate for the specific training data but was less able to "generalize".

c. The networks, when interfaced to an optimization procedure, were consistently able to extract improved designs. The trend was towards larger reductions in weight as the number of training pairs was increased. Weight reductions in excess of 4% over any information in the training data were achieved.

d. The configurational designs obtained for the various hidden-layer/IOP/E combinations were quite similar. The final configurations for the truss design were consistent.

The ten bar truss provided a simple example to describe some of the major goals of this research. It also provided a means to describe the neural network design space modeling procedure and implementation for design.

8.1.2 Three Spar Wing-Box

The second configurational design example was that of a three spar wing-box and was intended to represent a more involved and realistic design problem. The design parameters were the spar locations. Training data sets composed of 20 to 100 different wings were considered. The wings were designed using the FSD approach described previously and a modal analysis was performed for each of the resulting designs. A series of neural networks were considered. A second set of 100 wing designs were developed and then propagated through the neural networks to provide a means to determine the extrapolation ability of the various hidden layer/training pair combinations. A number of trends were noted:

a. The maximum error values (based on the 100 new designs) for nearly all network combinations showed a decline as the number of training pairs increased. This was the same behavior noted for the ten bar truss. Two of the nine combinations did not show this trend, however. It was possible that two designs fell near or in high gradient regions of the design space (i.e., small changes in the design variables result in large changes in the displacement and first natural frequency characteristics).

b. The RMS error levels consistently showed the expected error reduction with increasing IOP number. For both maximum and RMS errors, the most notable drop in error occurs between the 20 and 50 IOP cases. Error was typically reduced by 50% through the incorporation of the 50 IOP training sets.

c. Optimal configurations that were determined from the neural network/math-programming optimization procedure showed excellent agreement between the network predicted values and the actual FSD values for the corresponding designs. Most values were within 1% error. Considering the comparative computational efficiency between the neural network representation and the FSD procedure (typically at least an order of magnitude faster for the neural network), the neural network provided a more efficient, yet highly accurate, design alternative to the fully stressed design.

d. The neural network/math-programming procedure was able to extract improved designs for four of the five network outputs (weight, first, second, and third natural frequency). The tip displacement design was not lower but comparable to the minimum tip displacement present in the training data. The reductions obtained from the other four network outputs indicate the extrapolation ability of the neural network.

The wing-box provided a more involved application of neural networks to configurational design. An effective design space representation was obtained from the neural network and improved designs were ultimately determined.

8.2 Recursive Training For Continuous Design Variables

Recursive learning was investigated as a means to address the costly process of training data generation and to provide a rational approach to the selection of training information. In recursive learning, neural network training data was generated based on the neural network's own prediction of potentially desirable structural configurations.

8.2.1 Ten Bar Truss

The development of the ten bar truss's design space was demonstrated through a series of contour plots that displayed the truss FSD weight as a function of the coordinates of a single node. A number of observations were made:

a. The initial set of ten IOP's was insufficient to model the true nature of the design space. These initial ten designs were selected randomly in the allowable design region.

b. By the time that the IOP set had increased to 20 designs, a much more accurate representation of the design space was achieved. Two of the five recursive designs were situated near the global minimum weight location, while a third isolated a local minimum and two of the other recursive designs went to extremes of the design space. These last three recursive design locations, which are local minima, were attributed to the behavior of the math-programming optimization implementation.

c. By the point at which 40 designs were used, 25 random and 15 recursive, the nature of the design space was well described, and the majority of the local maxima and minima were well defined. Five of the ten recursive designs isolated and further refined the global minimum..

d. It was particularly interesting that a local minimum shown in the design space occurred in a region where no training data was present and yet a more detailed evaluation showed that there was indeed a local minimum in this location.

The ten bar truss described, in a graphical way, the progression of the design space development as the recursive learning procedure proceeded. The procedure was able to isolate the global minimum after only 20 designs were used to train the neural network. This is significantly less data than has been used in the previous applications, and indicated the promise and ability of the recursive learning procedure.

8.2.2 Four Spar Wing-Box

The four spar wing-box was meant to represent a more realistic and involved recursive learning example. Rather than simply consider weight as the sole objective function, all five network output quantities (W, δ_{TIP}, ω_1, ω_2, ω_3) were treated independently as the objective functions. This allowed for the recursive refinement of five specific regions of the four-dimensional design space. The major items to note from this application were:

a. Even after only a single recursive iteration the region of the design space that provides least weight was indicated. As the iterations progressed, the least weight design continued to improve. When compared to the designs obtained from larger, randomly selected training sets, it was seen that the recursively obtained least weight configuration was more than five percent lighter.

b. The recursively obtained designs for the tip displacement show the same characteristics noted for the least weight designs: a slight reduction in displacement as the iterations progress and a final recursive design that represented an improvement.

c. The limitation of the recursive procedure is that only specific regions of the design space become refined while the bulk of the design space may be less than adequately represented. This must be taken into account when approaches are developed to select new designs to add to the training set.

8.3 Material Selection with Discrete Design Variables

The material system design problems were posed as combinatorial optimization problems with discrete design variables. Neural networks were trained with the discrete material vectors and training data provided by the ASTROS system. The simulated annealing algorithm was used to extract least weight material systems from the neural network representations of the design spaces. Various neural network hidden layer geometries and IOP set sizes were considered. It was noted that the neural networks were able to represent the discrete information effectively, and the simulated annealing procedure was able to extract designs that significantly reduced the objective functions.

8.3.1 Ten Bar Truss

The ten bar truss example was formulated to provide a simple and descriptive problem involving discrete design variables and combinatorial optimization. Weight was considered as the objective function for the combinatorial optimization problem. A number of interesting results were obtained from this example:

a. For the four various hidden layer geometries that were considered, there were some material selections that were dominant regardless of the IOP level of training data. It was noted that as the hidden layer number increased, the material which had the highest specific stiffness was dominant. It appeared that as the number of neurons in the hidden layer increased, the ability of the neural network to extract improved designs also improved.

b. For each of the four networks as the IOP number increased the networks were able to provide better designs through simulated annealing. This was behavior that had been noted in previous continuous-valued design variable examples. The addition of new training data improved the design space representation and allowed for the extraction of an improved design.

c. The accuracy with which the neural networks predicted weight for the truss material systems also improved as the IOP number increased for certain networks. As yet unexplained, some of the larger networks actually showed a trend opposite to what was expected. This was directly opposite to the behavior that was noted for all the previous continuous design variable problems and indicates that the "ideal" geometry of the network is probably problem dependent.

d. The simulated annealing procedure was always able to isolate the global minimum weight material system as determined from an exhaustive search of the neural networks.

e. The simulated annealing demonstrated the ability to extract improved designs from the neural network representation of the design space. All network predictions, for all hidden layer geometries and IOP sizes, provided some reduction in weight over the least weight design present in the training data. The typical weight reductions for the higher IOP set networks were near 12%.

The ten bar truss design example was successful in verifying the neural network's ability to represent discrete valued design information, and the simulated annealing procedure was able to isolate the near-global minimum weight material combination for all the networks investigated.

8.3.2 ACOSS II Space Truss

The ACOSS II space truss material system design represented an extremely large combinatorial optimization problem, and was formulated to extend the limits of design space mapping using neural networks. The ability of the simulated annealing procedure to extract improved designs from such a large combinatorial set was also considered. This design study lead to a number of conclusions:

a. The neural network did not provide accurate weight values for material systems not present in the training data for cases where the design weight was appreciably lower than the lowest value in the training set. This was partially due to the training data scaling employed. Two linear scaling approaches were adopted but accurate extrapolation from the training data did not appear feasible. Though the weight estimates were of limited accuracy, weight trends in the design space appeared to be reasonably well defined.

b. It was noted that the ability of the simulated annealing procedure to extract improved designs was a strong function of the IOP number. Typical design weights obtained from the 50 IOP training sets for the various network geometries were above the

expected least weight design. As IOP number increased, the SA designs improved and the best case was only a fraction of a percent heavier than the expected least weight design. This was significant considering the large combinatorial size of the problem and the very small subset of possible material combinations used as training data for the design space formulation.

c. Increasing the number of neurons in the hidden layer did not significantly affect the designs extracted through simulated annealing. There did seem to be a slight increase in weight occurring as the number of neurons increases, an increase that was most pronounced at lower IOP sets.

d. The simulated annealing procedure was able to extract improved designs from every network geometry/IOP combination. This is a strong indication of the ability of the NN/SA procedure for large combinatorial design problems.

Without the foreknowledge about which material systems would provide improved designs, it would be virtually impossible to come up with an "optimal" material system for a problem this large. A random search of the design space, involving the design of a large number of ACOSS II trusses using randomly selected material systems could provide a marginal design at best. The neural network/simulated annealing approach provided the means to address this difficult combinatorial optimization problem.

8.3.3 Multi-Spar Composite Wing-Box

This problem represented a preliminary attempt to model the design of a composite wing skin as a combinatorial optimization problem. Though only a limited study was performed the following observations were made:

a. Alternative methods for representing the design variables in combinatorial optimization problems are possible. This was done with the "binary" representation in this problem for the individual lamina orientations.

b. Even a limited attempt to define the design space can help provide insight into important parameter trends. Though first natural frequency may not be the most appropriate measure of design performance, an "improved" design could be determined with limited training data.

8.4 Design for Survivability

The preliminary design of damage tolerant structures was presented as a special application of design space modeling with neural networks. Two distinctly different approaches and examples were considered. In both cases the purpose was to investigate the ability to effectively quantify the design space for further analysis.

8.4.1 Helicopter Tail-Boom

The first involved the configurational design of a helicopter tail-boom, in which six possible damage conditions were considered.

a. In this case it was possible to develop a model of the design space defined by least weight designs which had been subject to a series of damage conditions.

b. Though significant improvements were not achieved in the selected measures of merit, some design improvements were noted.

c. This type of representation of the characteristics of a design has been achieved and could be used as part of a "survivability" study to determine improved configurational designs.

8.4.2 Five Spar Wing-Box

In the second example, an undamaged baseline wing was designed for flutter and stress constraints. The neural network was then used to predict flutter occurrence and the natural frequencies for the complete set of possible damage conditions.

a. The neural network representations were able to determine the effects that damage conditions would have on the characteristics of the given structures.

b. Modeling the design space defined by various damage conditions requires enough detail in the design space so that an adequate quantitative representation can be achieved. In this case the ability to determine the influence of damage on flutter speed was hampered by the limited number of damaged designs which experienced flutter.

8.5 Conclusions

The research documented in this report has been intended to study the application of finite element based, structural analysis and design methods to the preliminary design of flight vehicle structures. An approach for the use of neural networks in quantifying the structural design space and the use of this design space representation in preliminary structural design has been developed. The design variables considered were both continuous and discrete in nature. A number of applications were considered ranging from configurational design to component material selection. In each case the goal was to allow for the use of detailed finite element representation of the structure at a point in the design process where the relative orientation of the structural components was not known or the materials not yet selected. This is in contrast to the conventional use of finite element models in the more detailed design phases.

The most obvious conclusion to be drawn from the work is that the "finite element modeling, analysis and optimization methods" must be both efficient and accurate *sources of design information.* Developing the information required to perform a finite element analysis of a given structural concept can be a very time-consuming process. Though powerful interactive and even graphical techniques are being developed to expedite this process, if these types of structural representations are to be used at the earliest stages of the preliminary structural design, they must be amenable to iterative or automated processing. One would not want to be in the position of eliminating a particular structural concept because it would take "too long" to assemble another finite element model.

Of even greater importance is the accuracy of the information which is provided by the analysis or optimization procedures. The accurate modeling of the design space is directly related to the accuracy of the analysis or optimization procedure. In a number of the cases presented in this report, the ASTROS system was used to provide candidate design concepts and design space information. ASTROS in a complex and sophisticated analysis/design program and the results which it provides must always be carefully evaluated. This is more difficult when repeated or automated processing of many candidate structural configurations are being considered. As additional constraints and disciplines are added to MDO methods, the possibility increases that problems related to modeling and analysis within a particular discipline will result in inaccurate results. The ability to quantify the design space and thus select between candidate designs will always be limited by the accuracy of the fundamental design information.

This research focused on the use of artificial neural networks as a means for the storing and processing of preliminary structural design information. In the past, neural networks have been used as alternatives for analysis in the sizing of structural components, not for the determination of structural geometry or the selection of materials as was done in this work.

Recursive training of the neural networks was considered in order to obtain accurate design space representations with as little training data as possible. The selective

development of the "costly" design information from the finite element analysis or MDO procedures appears to be an important consideration in the practical application of these methods to the preliminary structural design problem. Even with the development of automated modeling and pre-processing capabilities, the time spent in developing the information needed to characterize the design space for a particular structural concept will continue to be a great concern.

A series of applications involving discrete design variables in structural design were examined. Experience with discrete design variables, or combinatorial optimization problems, is somewhat limited in the area of structural design although there are many important issues in this area. This work has indicated the potential for design space mapping using the neural networks for this class of design problem. These approaches to modeling the design space can be combined with new developments in combinatorial optimization such as simulated annealing, to provide new opportunities to the structural designer.

Much of the effort in this work was devoted to providing experience with the development of neural networks to represent the structural design space. This was limited to the use of feed-forward networks with back-propagation training but useful insight was gained on the use of these networks in the structural design space modeling problem. The most consistent trend noted from the results was that increasing the amount of training information resulted in a reduction in error when the network results were compared to actual designs using the configurations obtained. This implies that the quality of the design space representation improves with increasing training set size. Both the RMS and maximum error values obtained for the configurational design examples displayed this trend. For the material system design examples, it was seen that the ability to extract improved designs improved significantly as the training set size was increased. The material system design results for the ACOSS II, which used discrete design variables and very sparse training sets (relative to the total combinatorial complexity of the problem), showed a consistently high percent error between the network predicted weight values and the actual ASTROS-designed weights for corresponding material systems. The trends in the discrete design space were, however, correctly represented, allowing for the extraction of significantly improved designs.

The observations related to training set size prompted the development of the recursive training procedure. The primary purpose was to provide a rational approach to the selection of the training information with the eventual goal of determining how to provide the same level of accuracy with less information. For the design problems considered, the recursive procedure was able to isolate and refine those regions of the design space where the optimal configurations were located. A relatively small number of iterations were required to isolate these regions. Some occurrences were noted where local minima were isolated, but this was a result of the optimization procedure implementation and not an inherent problem in the recursive procedure. The development of improved methods for selecting candidate designs to populate the design space appears warranted.

The application of neural networks for the representation of design information appears promising for a wide range of design related issues. There are several additional areas where more research should be considered.

1) Investigation into the use of neural networks for representing design spaces where both discrete and continuous design variables are simultaneously present should be considered. This would further enhance the ability to model realistic structural design problems at the conceptual or early preliminary design phase. The difficult issues that arise for this type of application involve determining the appropriate algorithms for extracting improved designs from such a mixed design variable optimization problem.

2) Further investigation into recursive training for both continuous and discrete design space representations would be useful. This has particularly strong promise for large dimension continuous design problems and problems where the combinatorial size is also large. Isolation of promising regions of the design space with as little training data generation as possible would be the goal.

3) Finally, additional effort is required to help establish requirements on the neural network geometries, or types, for appropriate design problems. Only a cursory evaluation of the influence of the network geometries was achieved in this effort. Since there is much current interest in the use of artificial neural networks, it is hoped that one can take advantage of new developments in this area and apply them to the problem of defining the design space for structural concepts.

9. REFERENCES

1. Ashley, H., "On Making Things the Best Aeronautical Uses of Optimization," Journal of Aircraft, Vol. 19, No. 1, AIAA 81-1738R, January 1982

2. Swift, R., Batill, S., "Application of Neural Networks to Preliminary Structural Design," AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics and Materials Conference,
April 1991

3. Johnson, E. and Venkayya, V., "Automated Structural Optimization System (ASTROS)," AFWAL-TR-88-3028 , December 1988

4. Schneider, G., and Godel, H., "Aeroelastic Considerations for Automatic Structural Design Procedures," Messerschmitt-Golkow-Blohm GMBH, Federal Republic of Germany, 1985, p. 1

5. McCulloch, W. S. and Pitts, W., "A Logical Calculus of the Ideas Imminent in Nervous Activity," Bulletin of Mathematical Biophysics, 5, 115-133, 1943

6. Hebb, D. O., The Organization of Behavior, John Wiley & Sons, New York, 1949

7. Rosenblatt, R., Principles of Neurodynamics, New York, Spartan Books, 1959

8. Swift, R.,"Structural Design Using Neural Networks," Ph.D. Dissertation, Department of Aerospace and Mechanical Engineering, University of Notre Dame, November 1992

9. Rehak, D., Thewalt, C. and Doo, L., "Neural Network Approaches in Structural Mechanics Computations," Computer Utilization in Structural Engineering (Proceedings of the ASCE Structures Congress of 1989)

10. Hajela, P., and Berke, L., "Neurobiological Models in Structural Analysis and Design," AIAA/ASME/ASCE/AHS/ASC 31st Structures, Structural Dynamics and Materials Conference, Part 1, pg. 345, April 1990

11. Berke, L. and Hajela, P., "Application of Artificial Neural Nets in Structural Mechanics," presented at the Lecture Series on "Shape and Layout Optimization of Structural Systems" at the International Center for Mechanical Sciences, Udine, Italy, July 16-20, 1990, NASA Technical Memorandum 102420

12. Swift, R. and Batill, S., "Simulated Annealing Utilizing Neural Networks for Discrete Design Variable Optimization Problems in Structural Design," 33rd AIAA/ASME/AHS/ASC Structures, Structural Dynamics and Materials Conference, April 1992, AIAA-92-2311

13. Swift, R, and Batill, S., "Damage Tolerant Design using Neural Networks," AIAA Aerospace Design Conference, February 1992, AIAA-92-1097

14. Swift, R. and Batill, S., "Structural Design Space Definition Using Neural Networks and a Reduced Knowledge Base," AIAA/AHS ASEE Aerospace Design Conference, February 1993, AIAA 93-1034

15. Carpenter, W. and Barthelemy, J., "Comparison of Polynomial Approximations and Artificial Neural Nets for Response Surfaces in Engineering Optimization," 33rd AIAA/ASME/AHS/ASC Structures, Structural Dynamics and Materials Conference, April 1992, AIAA-92-2247

16. Fu, B. and Hajela, P., "Minimizing Distortion in Truss Structures-A Hopfield Network Solution," 33rd AIAA/ASME/AHS/ASC Structures, Structural Dynamics and Materials Conference, April 1992, AIAA-92-2302

17. Linse, D. J. and Stengel, R. F., "Identification of Aerodynamic Coefficients Using Computational Neural Networks," 30th Aerospace Sciences Meeting & Exhibit, January 1992, AIAA-92-0172

18. Grandi, R.V, and Venkayya, V.B., "Structural Optimization with Frequency Constraints," Proceeding of the 28th AIAA Structures, Structural Dynamics and Materials Conference, April 1987

19. Swift, R., "Application of Finite Element Methods to the Preliminary Structural Design of Lifting Surfaces," M.S.A.E. Thesis, University of Notre Dame, 1989

20. MacNeil, R. H., "The NASTRAN Theoretical Manual," NASA SP-221(01), April 1971

21. Cook, R.D., Malkus, D.S., and Pelsha, M.E., Concepts and Applications of Finite Element Analysis, John Wiley & Sons, New York, Third Edition, 1989

22. Woodward, F. A., "An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow, Part I-Theory and Applications," NASA CR-2228, May 1973

23. Albano, E. and Rodden, W. P., "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows," AIAA Journal, Volume 7, February 1969, pp 279-285, and Volume 7, November 1969, p. 2192

24. Appa, K., "Constant Pressure panel Method for Supersonic Unsteady Airloads Analysis," Journal of Aircraft, Volume 24, October 1987, pp 696-702

25. Dwyer, W. J., "Finite Element Modeling and Optimization of Aerospace Structures," Technical Report AFFDL-TR-72-59, August 1972

26. Harder, R. L. and Desmarais, R. N., "Interpolation Using Surface Splines," Journal of Aircraft, Volume 9, No. 2, February 1972

27. Berke L., and Khot, N. S., " Use of Optimality Criteria Methods for Large Scale Systems," Technical Memorandum AFFDL-TM-74-70-FBR, April 1974

28. Klein, B., "Direct Use of External Principles in Solving Certain Optimizing Problems Involving Inequalities," Journal, Operations Research Society of America, Vol. 3, No. 2, May 1955, p. 168-175 and 548

29. Schmit, L. A., "Structural Design by Systematic Synthesis," Proceedings of the Second Conference on Electronic Computation, ASCE, September 1960

30. Gellatly, R. A. and Berke, L., "Optimal Structural Design," Technical Report AFFDL-TR-70-165, April 1971

31. Wilkinson, Lerner, and Taylor, "Practical Design of Minimum Weight Structures for Strength and Flutter Requirements," Journal of Aircraft, Volume 13, No. 1, June 1968

32. Lansing, W. et al., "Application of Fully-Stressed Design Procedures to Wing and Empennage Structures," Journal of Aircraft, Vol. 8, September 1971

33. Vanderplaats, G., "An Efficient Feasible Directions Algorithm for Design Synthesis," AIAA Journal, Volume 22, No. 11, November 1984, pp 1633-1640

34. Vanderplaats, G. and Moses, F., "Structural Optimization by Means of Feasible Directions," Journal of Computers and Structures, Volume 3, July 1973, pp 739-755

35. Gabriele, G. and Ragsdell, K., "Large Scale Nonlinear Programming Using the Generalized Reduced Gradient Method," Journal of Mechanical Design, Volume 102, No. 3, July 1980, pp 566-573

36. Baffes, P., "NETS User's Guide," COSMIC Program # MSC-21588 , Software Technology Branch, Lyndon B. Johnson Space Center, September 1989

37. Minsky, M. and Papert, S., Perceptrons, MIT Press, Cambridge

38. Werbos, P., "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Ph.D. Thesis, Harvard University

39. Hart, D.E., Hinton, G.E. and Williams, R.J.,"Learning Internal Representations by Error Propogation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol.1 Foundations D. E. Rumelhart and J.L. McClelland Editors, MIT Press, 1986

40. Parker, D. B., "Learning Logic," MIT Center for Computational Research in Economics and Management Science Technical Report TR-47

41. Miller, R., Walker, T. and Ryan, A., NEURAL NET APPLICATIONS AND PRODUCTS, SEAI Technical Publications, Madison, Georgia 1990

42. Hertz, J., Krogh, A. and Palmer, R., Introduction to the Theory of Neural Computation, 1991, Addison-Wesley Co., Redwood, California

43. Woods, D., "Back and Counter Propagation Aberrations," Proc. IEEE International Conference on Neural Networks, San Diego, July 1988, pp. I-473 to I-480

44. Rumelhart, D. E. and McClelland, J. L., Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press 1986

45. Schittkowski, K., Nonlinear programming codes, Lecture Notes in Economics and Mathematical Systems, pg. 183, Springer-Verlag, Berlin, Germany

46. IMSL Library Reference Manual, 1989, IMSL Inc., 7500 Bellaire Boulevard, Houston, TX 77036

47. Otten, R., van Ginneken, L., The Annealing Algorithm, Kluwer Academic Publishers, Boston, 1989

48. Kirkpatric, S., Gelatt, C.D., Vecchi, M. P, "Optimization by Simulated Annealing," *Science*, 220:671-680, 1983

49. Cerny, V., "Solving inverse problems by simulated annealing," July 1986, presented at the IBM workshop on statistical physics in engineering and biology, Lech.

50. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., "Equation of state calculations by fast computing machines," Journal of Chemical Physics, 21:1087-1092, 1953

51. Leong, H., "A new algorithm for gate matrix layout," Proceedings of the IEEE International Conference on Computer Aided Design, pages 316-319, Santa Clara, November 1986

52. Lundy, M., "Applications of the annealing algorithm to combinatorial problems in statistics," *Biometrika*, 72:191-198, 1985

53. van Laarhoven, P., "Theoretical and computational aspects of simulated annealing," *Ph.D. Thesis*, Erasamus Universiiet, Rotterdam, 1987

54. Carnevalli, P., Coletti, L., Patarnello, S., "Image processing by simulated annealing," *IBM Journal of Research and Development*, 29:569-579, 1985

55. Cerny, V., "Picture processing by statistically coupled processors: relaxed syntactical analysis," July 1986, presented at the IBM workshop on statistical physics in engineering and biology, Lech.

56. Smith, W., Barrett, E., Paxman, R., "Reconstruction of objects from coded images by simulated annealing," *Optic letters*, 8:199-201, 1983

57. Aarts, E., Korst, J., Simulated Annealing and Boltzmann Machines, Kluwer Academic Publishers, 1989

58. F. Hemmig, V. Venkayya, F. Eastep, "Flutter Speed Degradation of Damaged Optimized Flight Vehicles," AIAA/ASME/ASCE/AHS/ASC 20th Structures, Structural Dynamics and Materials Conference, April 1979

59. T. Erline, "Highly Survivable Truss Tail-Boom," Ballistic Research Labs, Aberdeen Proving Grounds, MD, November, 1978

60. E. Haug, J. Arora, Applied Optimal Design, John Wiley and Sons, New York, 1979, pg. 264-289