

NASA CONTRACTOR REPORT



NASA CR

0062546



TECH LIBRARY KAFB, NM

NASA CR-2607

STUDY OF FLUTTER RELATED COMPUTATIONAL PROCEDURES FOR MINIMUM WEIGHT STRUCTURAL SIZING OF ADVANCED AIRCRAFT

*R. F. O'Connell, H. J. Hassig,
and N. A. Radovcich*

Prepared by
LOCKHEED-CALIFORNIA COMPANY
Burbank, Calif. 91503
for Langley Research Center

LOAN COPY: RETURN TO
AFWL TECHNICAL LIBRARY
KIRTLAND AFB, N. M.



NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1976



0061546

1. Report No. NASA CR-2607		2. Government Accession No.		3.	
4. Title and Subtitle Study of Flutter Related Computational Procedures for Minimum Weight Structural Sizing of Advanced Aircraft				5. Report Date March 1976	
				6. Performing Organization Code	
7. Author(s) R. F. O'Connell, H. J. Hassig, and N. A. Radovcich				8. Performing Organization Report No. LR-26650	
9. Performing Organization Name and Address Lockheed-California Company P.O. Box 551 Burbank, California 91503				10. Work Unit No.	
				11. Contract or Grant No. NAS1-12121	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Contractor	
				14. Sponsoring Agency Code	
15. Supplementary Notes Technical Monitor: E. Carson Yates, Jr., Computer Aided Methods Branch, Structures & Dynamics Division, NASA Langley Research Center, Hampton, Virginia 23665 Final Report					
16. Abstract Results of a study towards the development of flutter modules applicable to automated structural design of advanced aircraft configurations, such as a supersonic transport, are presented. In this study, automated structural design is restricted to automated sizing of the elements of a given structural model. It includes a flutter optimization procedure; i.e., a procedure for arriving at a structure with minimum mass for satisfying flutter constraints. Methods of solving the flutter equation and computing the generalized aerodynamic force coefficients in the repetitive analysis environment of a flutter optimization procedure have been studied, and recommended approaches are presented. Five approaches to flutter optimization are explained in detail and compared. An approach to flutter optimization incorporating some of the methods discussed is presented. Problems related to flutter optimization in a realistic design environment are discussed and an integrated approach to the entire flutter task is presented. Recommendations for further investigations are made. Results of numerical evaluations, applying the five methods of flutter optimization to the same design task, are presented.					
17. Key Words (Suggested by Author(s)) Design for Flutter; Structural Optimization; Aeroelasticity			18. Distribution Statement Unclassified - Unlimited Subject Category 32; Structural Mechanics		
19. Security Class. (of this report) Unclassified	20. Security Class. (of this page) Unclassified	21. No. of Pages 127	22. Price* \$5.75		

SUMMARY

Results of a study towards the development of flutter modules applicable to automated structural design of advanced aircraft configurations, such as a supersonic transport, are presented. In this study automated structural design is restricted to automated sizing of the elements of a given structural model. It includes a flutter optimization procedure; i.e., a procedure for arriving at a structure with minimum mass for satisfying flutter constraints. Methods of solving the flutter equation and computing the generalized aerodynamic force coefficients in the repetitive analysis environment of a flutter optimization procedure have been studied and recommended approaches are presented. Five approaches to flutter optimization are explained in detail and compared. An approach to flutter optimization incorporating some of the methods discussed is presented. Problems related to flutter optimization in a realistic design environment are discussed and an integrated approach to the entire flutter task is presented. Recommendations for further investigations are made. Results of numerical evaluations, applying the five methods of flutter optimization to the same design task, are presented.

TABLE OF CONTENTS

SUMMARY	iii
TABLE OF CONTENTS	v
SYMBOLS AND DEFINITIONS	ix
1. INTRODUCTION	1
1.1 General	1
1.2 Objectives of Study	3
2. OVERVIEW OF THE FLUTTER OPTIMIZATION TASK	3
3. SOLUTION OF THE FLUTTER EQUATION	5
3.1 The Generalized Flutter Equation	5
3.2 Types of Solution Sought	6
3.3 Methods of Obtaining Point Solutions	8
3.3.1 Bhatia Method	8
3.3.2 Phoa Method	9
3.3.3 Lockheed Program 165	11
3.3.4 Desmarais-Bennett Method	11
3.3.5 Two-Dimensional Regula Falsi	12
3.3.6 Conclusion	13
3.4 Minimum Damping in Hump Mode	13
3.5 Recommendation	16
4. MODALIZATION	16
4.1 General	16
4.2 Types of Modes	17
4.3 Number of Modes	18
4.4 Updating of Modes	19
4.5 Recommendations	20
5. AERODYNAMICS	20
5.1 Introduction	20
5.2 General	21
5.2.1 Analytical Integration	23
5.2.2 Numerical Integration of the Product of Displacement and Pressure	23
5.2.3 Numerical Integration of the Pressures	24
5.2.4 Finite Element Approach	24
5.3 Basic Formulation	25

5.4	Factors Affecting The Efficiency of The Numerical Evaluation of The Matrix of Generalized Aerodynamic Force Coefficients	26
5.4.1	Matrix Population	27
5.4.2	Interpolation for Arbitrary k Value	27
5.4.3	Number of k Intervals	30
5.4.4	Sequence of Multiplications	31
5.4.5	Form of Inputting the Angle-of-Attack Generating Matrix	33
5.5	Summary of Comparisons	33
5.5.1	Input Storage Requirements	33
5.5.2	Core Space	33
5.5.3	Read-in	33
5.5.4	Number of Computational Operations	34
5.6	Conclusions and Recommendations	34
6.	METHODS OF OPTIMIZATION FOR FLUTTER	36
6.1	General	36
6.2	Rudisill-Bhatia Approach	36
6.2.1	Velocity Gradient Search	37
6.2.2	Mass Gradient Search	38
6.2.3	Gradient Projection Searches	40
6.2.4	Concluding Remarks	44
6.3	The Weight Gradient Method of Simodynes	44
6.3.1	The Method of Simodynes	44
6.3.2	Discussion of the Method	46
6.3.3	A Modification of the Method of Simodynes	47
6.3.4	Discussion of the Modified Method	48
6.3.5	Assessment of the Method	50
6.4	An Interior Penalty Function Method	50
6.4.1	Description of Method	50
6.4.2	Discussion of the Method	52
6.4.3	Assessment of the Method	54
6.5	A Method of Feasible Directions	55
6.5.1	Description of Method	55
6.5.2	Principal Characteristics of the Method	58
6.5.3	Assessment of the Method	61
6.6	An Optimization Method Using Incremented Flutter Analysis	62
6.6.1	Main Features	62
6.6.2	Present Form of Program	63
6.6.3	Concluding Remarks	67

6.7	Comparison of Optimization Methods	68
6.7.1	General	68
6.7.2	Arbitrary Step-Size Procedures	68
6.7.3	Defined Step-Size Procedures	69
6.7.4	Formulation of a Resizing Procedure	72
7.	CONSIDERATIONS RELATED TO A REALISTIC DESIGN ENVIRONMENT	73
7.1	Structural Model	73
7.2	Multiple Flutter Speed Constraints	77
7.3	Damping Constraints	78
7.4	Mass Ballast	81
7.5	Interface With Strength Optimization	82
8.	COMPUTATIONAL ASPECTS OF THE FLUTTER TASK	84
8.1	The Complete Flutter Task	85
8.1.1	Flutter Survey	85
8.1.2	Initial Structural Resizing	88
8.1.3	Flutter Optimization	92
8.2	Aspects of the Computing System	95
9.	CONCLUSIONS AND RECOMMENDATIONS	98
	APPENDIX A - NUMERICAL EXAMPLES OF RESIZING PROCEDURES	100
A.1	Introduction	100
A.2	Structural Model	101
A.3	Method of Simodynes	104
A.4	Gradient Methods of Rudisill and Bhatia	109
A.5	Interior Penalty Function Method	112
A.6	Method of Feasible Directions	113
A.7	An Optimization Method Using Incremented Flutter Analysis	114
	REFERENCES	115

SYMBOLS AND DEFINITIONS

$[]$	square, rectangular matrix
$[]^T$	transpose of a matrix
$\{ \}$	column matrix
$[]$	row matrix
$[]$	diagonal matrix
$[A(ik)]$	aerodynamics matrix (function of k and Mach number), modalized aerodynamics matrix
$A_{ij}, A_{ij}(k)$	generalized aerodynamic force coefficients
$[AIC], [AIC(k)]$	basic aerodynamics influence coefficients (function of k and Mach number) defined by equation (5.13)
a_n, a_{n+1}	amplitudes of successive cycles
C	arbitrary constant (equations (7.16), (8.1))
C_i	ratio between m_i and P_i : $m_i = C_i P_i$; elements of a basic resizing column (equation (6.53))
c	reference chord
$D()$	flutter determinant
$[D]$	viscous damping matrix
$[D_j]$	matrix relating control system displacements to structural displacements
$[DX]$	interpolation and differentiation matrix relating slopes at downwash collocation points to displacements at structural nodes (Section 5.3)
$[DZ]$	interpolation matrix relating translations at downwash collocation points to displacements at structural nodes (Section 5.3)
EAS	equivalent airspeed
EI	bending stiffness

$[F]$	combination of modal displacement matrices (equation (5.14))
f_i, f_j	general modes of displacement
g	structural damping, 2γ
GJ	torsional stiffness
$[H]$	interpolation matrix relating displacements at lumped aerodynamic load points to displacements at structural nodes (Section 5.3)
$H_j(p)$	transfer function of automatic control system (function of p)
$\{h\}$	column matrix of displacements at aerodynamic load points
h	constraint quantity (equation (6.43))
$[HAW]$	unmodalized aerodynamics matrix (function of k and Mach number) (equation (5.15))
$[K]$	stiffness matrix, modalized stiffness matrix
$[K_0]$	base stiffness matrix
$[\Delta K_i]$	incremental stiffness matrix per unit design variable i .
\bar{K}	constant (equation (6.22))
K_1	constant (equation (6.21))
k	reduced frequency $k = \frac{\omega c}{V}$
KEAS	knots equivalent airspeed
$\mathcal{L}_\ell(k)$	polynomial multipliers used in Lagrange's interpolation formula
$[M]$	mass matrix, modalized mass matrix
$[M_0]$	base mass matrix
$[\Delta M_i]$	incremental mass matrix per unit design variable i
M	total mass associated with the design variables
m_i	design variable associated with structural mass (in mass or weight units)
$\{\Delta m_i\}$	resizing column

P_i	design variable of Reference 14
$P(m_i, r)$	modified objective function (Section 6.4.1)
$[PKI]$	pressure-kernel integral matrix (equation (5.4))
p	complex root of flutter equation for a given flight condition ($p = (\gamma + i)k$)
p_j	aerodynamic lifting pressure distribution corresponding to deflection mode j
p^n	aerodynamic lifting pressure distribution mode
q	modal degrees of freedom (modal participation coefficients)
$\{q\}$	modal column corresponding to solution of characteristic flutter equation
r	penalty function weighting factor
$[r]$	modal row corresponding to solution of characteristic flutter equation
V	speed, flutter speed
V_f	flutter speed
V_{mc}	most critical flutter speed
V_R	required flutter speed (minimum allowable flutter speed: 1.20 V_D for commercial, 1.15 V_L for military)
V_D	design speed according to Federal Aviation Regulations
V_L	design speed according to military specifications MIL-A-008870A (USAF)
W	total weight associated with the design variables
W_0	total weight of base structure
W_1	arbitrarily chosen total weight reduction
W_2	arbitrarily chosen total weight associated with positive component of resizing vector
$[W], [W(k)] = [DX] + ik[DZ]$	angle-of-attack generating matrix (function of k) (Section 5.3)

$[WF], [WF]$	weighting matrix
$[WFD]$	differentiating and weighting matrix
x	coordinate in a fore-and-aft direction
y	coordinate in a lateral direction
$\{z\}$	column matrix of displacements of structural nodes
$[\bar{z}]$	matrix of modal columns of displacements of structural nodes
α	angle of attack, parameter in one-dimensional minimization
β_i	general design variable
γ	normalized real part of $p = (\gamma + i)k$
$\gamma_{\text{hump top}}$	maximum value of γ in hump mode
λ_1	constant (equation (6.10))
ρ	air density
φ	aerodynamic velocity potential
ω	circular frequency, rad/s
'	indicates derivative of a one-variable function
in-flight mode:	modal column corresponding to a characteristic solution of the flutter equation
flutter mode:	in-flight mode that becomes unstable within the velocity range considered
hump mode:	in-flight mode with a minimum damping point within the velocity range considered

STUDY OF FLUTTER RELATED COMPUTATIONAL
PROCEDURES FOR MINIMUM WEIGHT STRUCTURAL
SIZING OF ADVANCED AIRCRAFT

R. F. O'Connell, H. J. Hassig and N. A. Radovcich

Lockheed-California Company

Burbank, California

1. INTRODUCTION

1.1 General

One of the factors contributing to the profitability of an airplane is its payload/range capability. Given certain safety and performance requirements, there is a direct trade-off between structural weight and payload, and it is the ideal of each airplane designer to reduce structural weight. Although the ideal minimum weight design may be expensive to produce, overshadowing any payload/range gains, it provides a good starting point for a practicable design and a good basis for comparing different designs.

Structural weight minimization, of course, is not a new idea. It is one of the airplane designer's most critical tasks. It now has come to the forefront as a result of two developments.

First, it has become evident to the structural design engineer that the combination of finite element modeling, high speed computer capacity, and mathematical techniques makes it practicable to do detailed structural synthesis aimed at minimizing weight.

Second, the need for a comprehensive and detailed approach to structural design optimization has significantly increased with the advent of the supersonic transport. This follows from the fact that for a supersonic transport the return in terms of payload/range per pound of structural weight saved is considerably larger than for a subsonic transport. For instance, a one percent structural weight saving on a typical subsonic transport might result in an increased payload capability of one to two percent; on an arrow wing supersonic transport, recently studied by the Lockheed-California Company, a one percent structural weight saving resulted in a four percent increase in payload capability for the design range.

The subject of this report is flutter optimization; i.e., structural weight minimization with flutter constraints. The need for a systematic, possibly automated, approach to flutter optimization also has increased significantly. Subsonic transports, as they are known, and transonic transports,

as shown in artist's sketches, can be represented by simple, beam-type structural models that are satisfactory for optimization with flutter constraints. Flutter optimization for such designs can be done, and has been done, with available methods. The supersonic transports that are flying and those being studied, however, all have lifting surfaces that cannot be represented satisfactorily by simple beam-type structural models. This fact alone makes the task of flutter optimization an order of magnitude more complicated.

Although ad hoc approaches to flutter optimization still could lead to a satisfactorily optimized supersonic design, refined methods that take full advantage of the capabilities of the present computers, in regard to automation as well as interaction with the engineer, become attractive and possibly mandatory. This is especially true in view of the rapidly increasing capability for fast analysis and synthesis in the areas of structural modeling and analysis, stress optimization, and performance analysis supported by improved configuration control. Flutter optimization must keep abreast of these developments. A balanced improvement in capability in all disciplines will make possible, within a practicable time span, true in-depth comparisons between a large number of candidate designs.

The preceding paragraphs present generally well known justification for a concerted effort in improving methods of structural optimization with flutter constraints. Work performed during the subject study is part of such an effort.

Work towards the goal of a generally available automated or semi-automated structural optimization system, that includes items such as optimization for stress, flutter and controllability, multiple flutter speed and modal damping constraints, is still in a state of development. The present study has contributed to this goal in the following areas. Methods of computing the aerodynamics parameters to be used in a flutter optimization program have been compared in detail with respect to characteristics which are independent of a specific aerodynamics theory. A method for efficiently and reliably solving the flutter equation for roots of interest in a flutter optimization module has been developed. Five methods of flutter optimization have been compared in detail and the mechanics of the optimization process have been examined; numerical examples with all five methods have been generated for the same aircraft design. Recommendations for further study and for the design of a flutter optimization module have been made.

The principal results of this study are presented in this report. Background discussions and supporting material are presented in a companion report (Reference 1).

1.2 Objectives of Study

The objectives of this study are:

1. To survey and evaluate methods of representing unsteady aerodynamics parameters and make recommendations for a general, accurate and efficient formulation that minimizes the computational effort during the optimization process. The assessment of aerodynamics theories, however, falls outside the scope of this study.
2. To survey and evaluate methods of determining the flutter characteristics and make recommendations for a method that is reliable and efficient, and suitable for the optimization process.
3. To evaluate and compare a number of methods of structural optimization with flutter constraints and make recommendations for further evaluation in a realistic design environment.
4. To make preliminary recommendations for the design of a flutter optimization module.

2. OVERVIEW OF THE FLUTTER OPTIMIZATION TASK

Structural optimization with flutter constraints is both an extension of the structural optimization task related to strength and an extension of the flutter analysis task. Being an extension of two tasks that traditionally are considered to belong to different disciplines, flutter optimization must take into account requirements of both disciplines. Structural optimization requires that a structural model is used that incorporates sufficient structural detail, in terms of distribution of structural material, to aid the designer in defining hardware. Similarly the flutter analysis that is incorporated in the optimization process must be of an accuracy comparable to that used outside of flutter optimization. The latter refers to methods of representing the unsteady aerodynamics and methods of solving the flutter equation, since the more detailed a structural model is, the more accurate, from an idealized theoretical point of view, is the flutter analysis. From a practical point of view, structural sizing for strength requires more detail in the structural model than is required for adequate prediction of flutter characteristics.

Thus, the flutter optimization task starts with the definition of the structural model. This is one of the most crucial aspects of flutter optimization, and it involves a serious conflict between simplicity of approach and computer cost. Present computer technology, or methods of structural analysis, or both, may not permit a structural model with sufficient detail for a stress analysis to be used in flutter optimization; computer cost could be exorbitant due to the repetition of operations during the design process. Section 7.1

deals with this problem in more detail. Suffice here that associated with the choice of structural model is the selection of a practicable number of degrees of freedom for the vibration analysis that has to provide the modes for the modal reduction of the flutter equation, which is usually required to limit computer cost. If the degrees of freedom for the vibration analysis are a subset of the degrees of freedom of the structural model, complications arise if a nonlinear relationship between the stiffness matrix and the design variables results (see Section 7.1).

Without serious restriction on scope or accuracy of the analysis the mass matrix can be assumed to be a linear function of design variables. It is the sum of a basic matrix and as many elementary matrices as there are design variables associated with a mass change, each proportional to a design variable.

During the flutter optimization there is repeated need for determining roots of the flutter equation, each time that a structure that has undergone a resizing since the previous solution of the flutter equation. For many, if not all, of these solutions a remodalization is necessary based on vibration modes of the current configuration. It is found that for the optimization process to provide reliable, converged results, consistent with the capability of the structural model, more modal degrees of freedom are required in the flutter equation than for a routine flutter analysis (see Section 4).

Incorporation of state-of-the-art level aerodynamics in the flutter optimization process does not provide significant problems beyond those encountered in the usual flutter analysis. For a given external geometry the basic aerodynamics formulation is invariant with structural changes. The repetitive formation of generalized aerodynamic forces for successive, updated modalization of the flutter equation is simple and relatively inexpensive.

In view of the objectives and the scope of the present program, this report devotes major sections to important aspects of the flutter optimization procedure. Section 3 deals with the solution of the flutter equation. Section 4 deals with modalization. Section 5 presents part of considerable work devoted to the aerodynamics, with the remainder being presented in Reference 1. In Section 6, methods of optimization for flutter, evaluated during this study, are discussed. Numerical results obtained by applying these methods to a simplified optimization task are presented in Appendix A.

Against the background provided by these sections, Section 7 presents discussions of several additional problems and considerations that need to be studied in order to choose a rational approach for formulating a flutter optimization module.

In Section 8, computational aspects of the complete flutter task are delineated. This task includes flutter analysis as well as structural synthesis of a design that satisfies the flutter requirements.

Section 9 summarizes the conclusions of the present study and presents recommendations for future work.

3. SOLUTION OF THE FLUTTER EQUATION

3.1 The Generalized Flutter Equation

When using the k method the flutter equation can be written as:

$$\left[-\frac{1}{c^2} [M] k^2 + \frac{1+ig}{v^2} [K] - \frac{1}{2} \rho [A(ik)] \right] \{q\} = 0 \quad (3.1)$$

One of several possible methods of solving this equation is to determine the characteristic value $\lambda = \frac{1+ig}{v^2}$ for several values of the reduced frequency $k = \frac{\omega c}{v}$, keeping all other quantities in the equation constant (Reference 2).

In the p - k method the flutter equation is

$$\left[\frac{v^2}{c^2} [M] p^2 + (1+ig) [K] - \frac{1}{2} \rho v^2 [A(ik)] \right] \{q\} = 0 \quad (3.2)$$

and solutions $p = (\gamma + i)k$ are sought for selected combinations of values of v and ρ (Reference 3). The p - k method formulation is convenient for the inclusion of viscous damping and control system transfer functions. This is accomplished by writing:

$$\left[\frac{v^2}{c^2} [M] p^2 + \frac{v}{c} [D] p + (1+ig) [K] - \frac{1}{2} \rho v^2 [A(ik)] - \sum H_j(p) [D_j] \right] \{q\} = 0 \quad (3.3)$$

where $H_j(p)$, $j=1,2,\dots$, represents transfer functions of the control system and $[D_j]$ relates the control system displacements to the structural displacements; $[D]$ is a viscous damping matrix (Reference 3).

A further generalization of the flutter equation can be made by making the stiffness matrix and the inertia matrix functions of design variables

m_i , which is the standard procedure for structural optimization. In addition, other quantities, such as $[D]$, $[D_j]$, as well as transfer function coefficients in $H_j(p)$, may be made functions of design variables.

Equation (3.3) implies that the determinant of the square matrix on the left hand side is zero and thus, in a very general form, the characteristic equation corresponding to the flutter equation can be written as:

$$D \left\{ (\gamma+i)k, g, V, \rho, m_i \right\} = 0 \quad (3.4)$$

D is called the flutter determinant. For arbitrary values of the variables it has a complex value. Thus equation (3.4) represents two equations and, in principle, can be solved for two unknowns for given values of the other variables.

Letting $\gamma=0$ and solving for g and V corresponds to the traditional k method of solving the flutter equation. Solving for γ and k corresponds to the p - k method. Letting $\gamma=0$ and solving for k and V leads directly to the flutter speed for a given value of the structural damping, g .

Solving equation (3.4) for k and one of the design variables, assuming all other variables fixed, is a new use of the flutter equation. It is called Incremented Flutter Analysis (References 4 and 1), applications of which are included in Sections 6.2.3 and 6.6.

3.2 Types of Solution Sought

A flutter analysis in the traditional sense is the determination of the flutter characteristics of a given structure. It includes the calculation of any flutter speed that may occur at speeds up to or somewhat beyond a speed corresponding to the required flutter margin. It also includes the gaining of insight in the variation of frequency and aerodynamic damping at speeds below the flutter speed for several in-flight vibration modes of interest. Consequently, sufficient modal solutions are obtained for the construction of f - g - V diagrams (Figure 3-1) for several flight conditions.

A procedure for structural optimization with flutter constraints will most likely start with such a survey-type analysis. However, during the process of repeated resizing, leading to the optimum design, there is no need for determining complete f - g - V diagrams at each resizing; only point solutions are required. Point solutions found in the literature are of two types: 1) directly solving for the flutter speed (the combination k, V in equation (3.4)), and 2) determining the value of one design variable necessary to satisfy a given flutter speed constraint (the combination k, m_j in equation (3.4) where m_j is one of the design variables m_i).

ARROW WING SST CONFIGURATION - CONTRACT NAS1-12288

Symmetric Flutter Analysis - 20 Vibration Modes
 Two Rigid Body Modes Not Shown
 Mach Number = 0.6 Weight = 145,600 kg

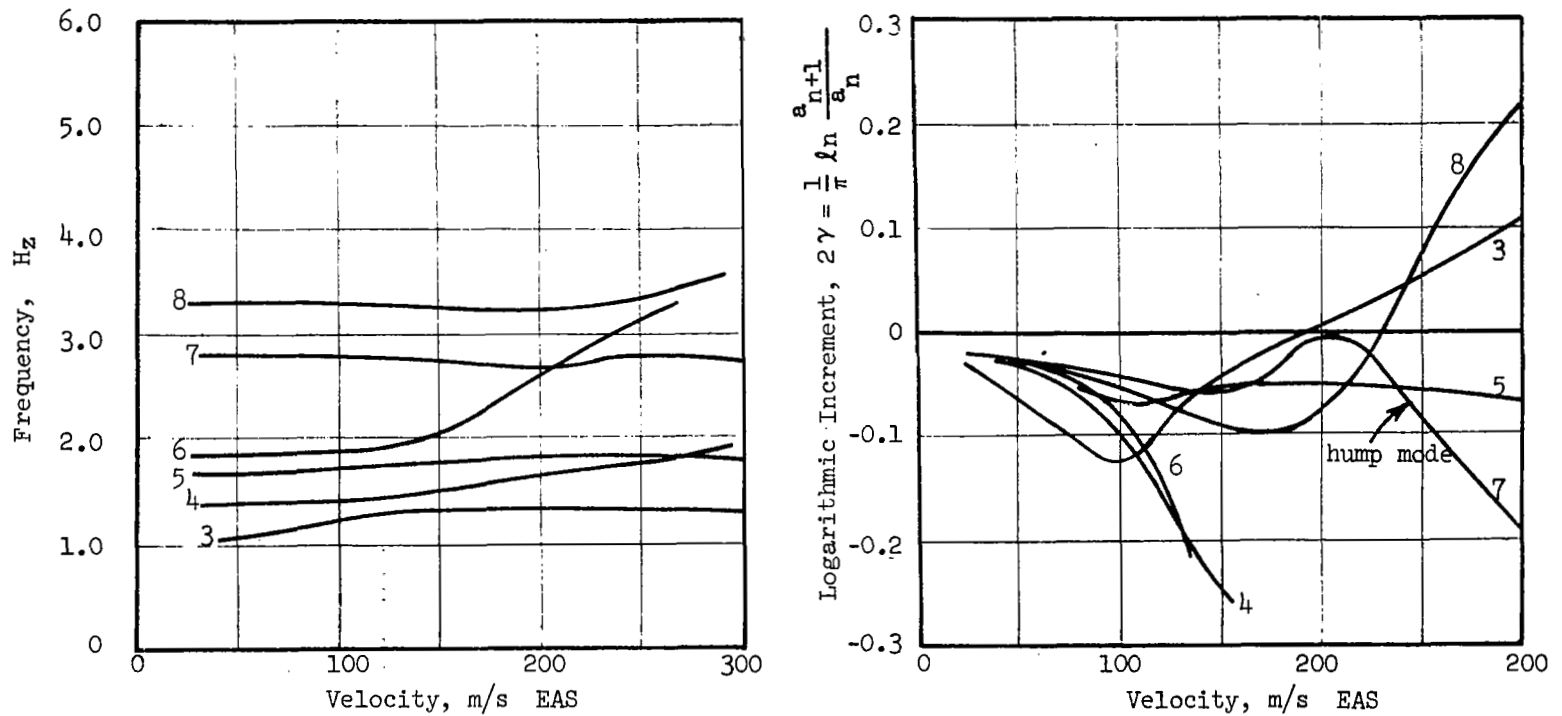


Figure 3-1: Example of Complete f-g-V Diagram

One additional type of point solution has been formulated during this contract, resulting in the determination of the minimum damping point of an in-flight mode. Such a point, if it exists, is of interest if the minimum damping point lies within the speed range considered. The associated mode is called a hump mode (See Figure 3-1). This point solution requires the solution of equation (3.4) and the equation

$$\frac{\partial \gamma}{\partial V} = 0 \quad (3.5)$$

for the three unknowns k , γ and V . Details of the formulation are given in Section 3.4. No numerical evaluation of the method has been made thus far.

The following section deals mainly with methods of obtaining point solutions for the flutter speed. It should be kept in mind that when such solutions are needed in an optimization program a solution for a similar structural configuration is usually available as a first approximation to the required solution.

3.3 Methods of Obtaining Point Solutions

Several methods for obtaining point solutions have been considered and evaluated to various depths. Their apparent efficiency, in terms of computational effort, is an important part of the evaluation. However, the degree of certainty with which a desired solution can be found is even more important.

The latter consideration refers to convergence problems and to problems associated with relating modal solutions at one value of V or k to modal solutions at another value of V or k .

The results of evaluations of the following methods are presented:

- Bhatia method (Reference 5)
- Phoa - Boeing method (Reference 6)
- Lockheed's Program 165 (p-k method, Reference 3)
- Desmarais-Bennett method (Reference 7)
- Two Dimensional Regula Falsi and Newton Raphson (Reference 8)

3.3.1 Bhatia Method - In Reference 5 Bhatia presents a method of solving directly for the flutter speed. Numerical evaluations of the method have been performed using data from the arrow wing study that Lockheed has conducted under contract NAS1-12288.

In Bhatia's method, which is based on the k-method approach, the structural damping, g , required for neutral stability is computed as a function of $1/k = \frac{V}{\omega c}$ by means of a Laguerre type extrapolation. It is an iterative

method that is initiated by choosing a trial value $1/k$ and computing the associated value of g and its first and second derivative with respect to $1/k$. The Laguerre extrapolation leads to a first approximation of the value of $1/k$ for which $g=0$. The process is repeated for this new value of $1/k$ until convergence is reached.

The method as presently programmed uses only aerodynamic matrices at preselected values of k , requiring a large number of preselected k values. The method also requires inputting the first and second derivatives of all aerodynamics matrices with respect to $1/k$. The method could be improved by using interpolation with respect to k to determine the aerodynamics matrix and its derivatives at arbitrary values of k from matrices given at a moderate number of preselected k values. Care must be taken that the interpolated results are defined uniquely over the range of k of interest for a particular solution to prevent the solution from oscillating between two values ("hunting").

Numerical evaluations were performed as part of this study. Difficulties were encountered in tracking the proper mode and in converging on the lower flutter speed of a hump mode. There is uncertainty whether the program can be modified such that the proper modal solution is always obtained.

At each step in the iteration towards the solution a characteristic value problem must be solved. This may prove to be costly in terms of CPU time.

3.3.2 Phoa Method - In Reference 6, Phoa presents a formulation of the flutter equation from a controls theory point of view. Although it is recognized that controls theory could prove to be of assistance in interpreting the flutter phenomenon, in the case of Reference 6 it leads to an equation that is essentially the same as equation (3.4). Phoa's method, based on the k -method approach, is in use at the Boeing Company. Discussions with Boeing personnel indicate that in the actual application the equation $\{D(\omega, V) - 1\} = \bar{D}(\omega, V) = -1$ is solved for V and ω .

The solution is accomplished in two steps. Constant velocity lines in the complex plane representing $\bar{D}(\omega, V)$ are intersected with the real axis. The values of the real parts at the intersections, as a function of the velocity, are used to determine an estimate of the velocity for which the real part of $\bar{D}(\omega, V)$ equals -1 . In an iterative process the accuracy of the solution is improved.

In numerical evaluation of this approach it was shown that the constant velocity lines may have two intersections with the real $\bar{D}(\omega, V)$ axis; this can be a source of problems (Figure 3-2).

The method is a sequence of two interpolations requiring many determinant evaluations. It is expected that very few, possibly not more than two or three, steps in the iteration process are required.

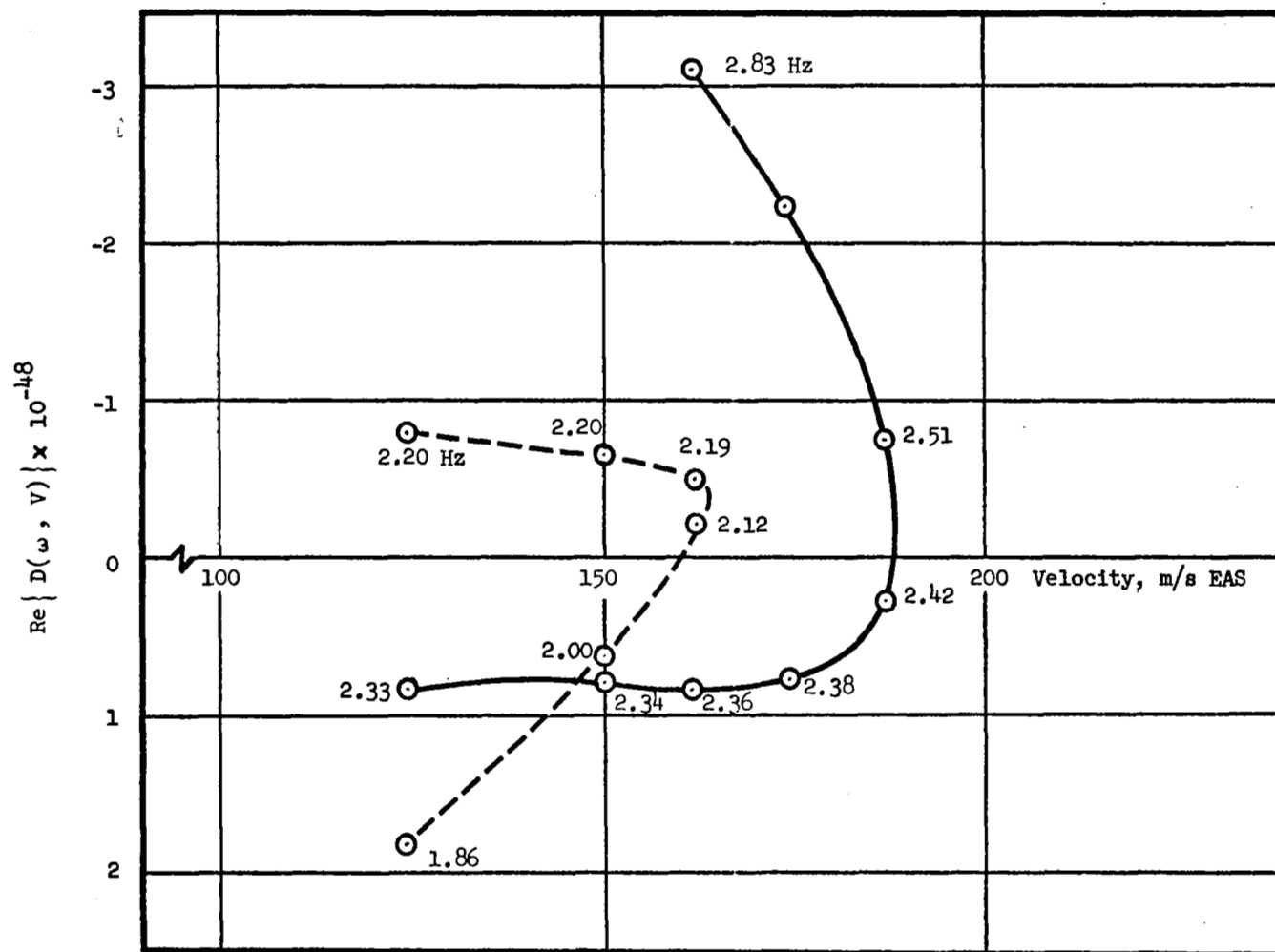


Figure 3-2: Value of $\text{Re}\{D(\omega, V)\}$ Corresponding to $\text{Im}\{D(\omega, V)\} = 0$ as a Function of Velocity

3.3.3 Lockheed Program 165 - Program 165 of Lockheed's Flutter and Matrix Algebra System (FAMAS) is based on the p-k method approach. It is designed to generate many point solutions, associated with in-flight modes, so that complete f-g-V diagrams can be constructed. The program solves equation (3,3) for $p=(\gamma+i)k$ given an initial trial solution. For determining the flutter speed, γ is evaluated at several values of the velocity. Flutter occurs at the speed for which $\gamma=0$.

The program has been used successfully in nonautomated numerical evaluations during this contract. Automation should be relatively simple and could be based on the following steps. At the estimated flutter speed an estimated frequency is used to start the process. Both are obtained from the solution for a previous structural configuration. Determinant iteration (see Reference 3) will lead to the actual value of γ at the estimated flutter speed. At a slightly perturbed velocity, using the damping and frequency already found as trials, γ is again evaluated. The two pairs of V and γ values thus found are used to initiate a One-Dimensional Regula Falsi procedure that leads to a value of V for which $\gamma=0$. The approach is expected to be quite efficient, except for the problem of assuring that subsequent solutions belong to the same in-flight mode.

3.3.4 Desmarais-Bennett Method - Reference 7 presents a fast and economical automated procedure to generate f-g-V diagrams, including the proper connection of point solutions of the flutter equation. The procedure is based on the k-method approach.

Reference 7 shows that the method is quite powerful in properly connecting point solutions. The sample cases in Reference 7, however, are obtained by partial deflation of the flutter determinant after each modal solution is found. Thus using this method would require solving for more roots than are of interest if only the flutter speed is required. Or, alternatively, if only the root of interest is determined, there is uncertainty whether the method will be as successful in following modal solutions as shown in Reference 7.

Application of this method to directly solving for the flutter speed could be programmed according to the following procedure.

The known solution for a base configuration is considered a reasonable estimate of the solution for a slightly modified configuration. Two k values, closely spaced according to the Desmarais-Bennett approach, are chosen such that flutter is expected to occur at a lower k value. Modal solutions at these two k values are obtained. The repeated sequence of linear extrapolation to the next k value and the Laguerre iteration described in Reference 7 is performed for the mode that is expected to give the flutter crossing and one or more additional modes on each side of this mode in the frequency spectrum. The additional modes are included to assure that a flutter crossing is obtained, in the event that an error in judgment is made in selecting the prime candidate mode for a flutter crossing.

The preceding conceptual evaluation defines the problems that need to be resolved when adjusting the Desmarais-Bennett method for use in a flutter optimization program and no numerical evaluation was considered necessary.

3.3.5 Two-Dimensional Regula Falsi - The concept of solving the two equations implied by equation (3.4) for two unknowns is not new. However, using this concept for directly solving the flutter equation for the flutter speed is relatively new. The need for such a solution arose with the advent of structural optimization with flutter speed constraints and, to the knowledge of the present authors, the first published record of solving directly for the flutter speed is Reference 9.

In that Reference the Newton-Raphson approach is used in two dimensions to determine flutter speed and, as a byproduct, flutter frequency. The Newton-Raphson approach is based on determining the value of a function and its derivatives for an initial set of trial values and extrapolating linearly to an estimate of the solution. In Reference 9, the derivatives are determined by a finite difference technique. The Two-Dimensional Regula Falsi approach uses three trial sets of the unknowns to construct two planes. The common point between those planes and the plane $D(\omega, V)=0$ defines the next estimate of the solution.

Table 3-1 compares the essential characteristic of the two methods. In the Newton-Raphson method with analytical evaluation of the derivatives, the formation of two derivative matrices is time consuming. In all methods the determinant evaluations are the most time consuming. Other operations, related to solving two linear equations with two unknowns, are trivial. Provisions to assure convergence are comparable for the two methods. Numerical experience with the Two-Dimensional Regula Falsi has indicated that problems with convergence on a solution are more easily solved than with the Newton-Raphson approach. It is concluded that the Two-Dimensional Regula Falsi approach is the more preferable one of the two.

It should be noted that both methods can be used for combinations of unknowns other than frequency and flutter speed. The Two-Dimensional Regula Falsi has been used successfully for solving for the value of one design variable, required to meet a given flutter speed, and the associated frequency. The method does not require the computation of derivatives. No interpolation or extrapolation of converged solutions is required, unless nonconvergence is encountered and an intermediate configuration is analyzed to assist in obtaining a better initial estimate of the solution for the configuration for which the original nonconvergence occurred. Finally, the solution sought is a combination of real values of the unknowns, rather than a series of complex modal solutions associated with in-flight modes. The equivalent of converging on the wrong mode, as may occur in seeking modal solutions, usually leads to nonconvergence, and a recovery procedure that is described in Reference 1. Thus mode switching to a non-flutter mode does not occur or, at worst, leads to nonconvergence. The chance of converging on the wrong flutter speed and frequency would seem to be quite small in view of the relatively small number of solutions within the region of interest of the unknowns. It has never occurred in the many test cases that have been run during this study.

TABLE 3-1. COMPARISON OF NEWTON-RAPHSON METHOD AND
TWO-DIMENSIONAL REGULA FALSI METHOD

Operation			Newton-Raphson		Two-Dimensional Regula Falsi
			Analytical Derivatives	Finite Differences	
Number of initial estimates			1	1	3
Interpolation of aerodynamics matrix required?			Yes	Yes	Yes
Derivative of aerodynamics matrix required?			Yes	No	No
Formation of derivative matrices required?			Yes	Trivial	No
Number of complex determinant evaluations per iterative step	First step	Value of determinant	1	3	3
		Derivative	2	0	0
	Each fol- lowing step	Value of determinant	1	3	1
		Derivative	2	0	0

3.3.6 Conclusion - On the basis of overall engineering evaluation, supported by numerical experience with all methods except the Desmarais-Bennett and Newton-Raphson approach, the Two-Dimensional Regula Falsi approach was considered most promising and chosen for further development (see Reference 1).

3.4 Minimum Damping in Hump Mode

Sufficient modal damping within the speed envelope can be assured by requiring sufficient damping in all modes at "all" speeds below the minimum required flutter speed or by requiring that the minimum damping in each mode, in so far as it occurs below the minimum required flutter speed, is equal to or larger than a given value.

To initiate exploration of the latter approach a method to determine the minimum damping in a hump mode was formulated.

The point of minimum damping in the hump mode is defined by the condition $\frac{\partial \gamma}{\partial V} = 0$, where γ defines the real part of the flutter root, $p = (\gamma + i)k$, in terms of the reduced frequency k . The quantity γ is a form of the logarithmic increment:

$$\gamma = \frac{1}{2\pi} \ln \frac{a_{n+1}}{a_n} \quad (3.6)$$

where a_n and a_{n+1} are amplitudes of successive cycles.

An expression for $\frac{\partial \gamma}{\partial V}$ is found as follows.

Consider the p - k method formulation of the flutter equation (equation (3.2)) and take the derivative with respect to V :

$$\begin{aligned} & \left[\frac{2V}{c} [M] p^2 + 2 \frac{V^2}{c^2} [M] p \frac{\partial p}{\partial V} - \rho V [A(ik)] - \frac{1}{2} \rho V^2 \frac{\partial}{\partial k} [A(ik)] \frac{\partial k}{\partial V} \right] \{q\} + \\ & + \left[\frac{V^2}{c^2} [M] p^2 + (1+ig) [K] - \frac{1}{2} \rho V^2 [A(ik)] \right] \left\{ \frac{\partial q}{\partial V} \right\} = 0 \end{aligned} \quad (3.7)$$

Choose a velocity V_1 for which $\frac{\partial \gamma}{\partial V}$ is estimated to be equal to zero. The solution of the flutter equation at V_1 is: $p = p_1$, $\{q\} = \{q_1\}$ and the characteristic vector of the transposed equation: $\{r\} = \{r_1\}$. Substituting this solution into equation (3.7) gives:

$$\begin{aligned} & \frac{2V_1}{c} [r_1] [M] \{q_1\} p_1^2 + 2 \frac{V_1^2}{c^2} [r_1] [M] \{q_1\} p_1 \frac{\partial p}{\partial V} + \\ & - \rho V_1 [r_1] [A(ik)] \{q_1\} + \frac{1}{2} \rho V_1^2 [r_1] [A'(ik_1)] \{q_1\} \frac{\partial k}{\partial V} = 0 \end{aligned} \quad (3.8)$$

where $[A'(ik_1)] = \frac{\partial}{\partial k} [A(ik)]$ evaluated at $k = k_1$.

With $p = (\gamma + i)k$:

$$\frac{\partial p}{\partial V} = \frac{\partial \gamma}{\partial V} k_1 + (\gamma_1 + i) \frac{\partial k}{\partial V} \quad (3.9)$$

Substituting equation (3.9) into equation (3.8) leads to a complex equation, and thus two equations in the two unknowns $\frac{\partial \gamma}{\partial V}$ and $\frac{\partial k}{\partial V}$ from which $\left(\frac{\partial \gamma}{\partial V}\right)_1$ can be determined.

The process can be repeated for V_2 , leading to $\left(\frac{\partial \gamma}{\partial V}\right)_2$. A one-dimensional Regula Falsi approach will lead to the value of V for which $\frac{\partial \gamma}{\partial V} = 0$.

In the above approach two characteristic value problems must be solved for determining the first iterated value of V for $\frac{\partial \gamma}{\partial V} = 0$. Each following step requires solution of one characteristic value problem.

It should be noted that damping versus speed curves may be rather flat and for practical purposes a converged value of γ may not define a converged value of V . This causes no problem since the most likely application of these procedures is in connection with an inequality constraint such as:

$$\gamma_{\text{hump top}} \leq \gamma_{\text{max allowed}} \quad (3.10)$$

Determining the minimum damping in a hump mode can be combined with solving for the value of a design variable satisfying the constraint:

$$\gamma_{\text{hump top}} = \gamma_{\text{max allowed}} = \bar{\gamma} \quad (3.11)$$

For $V = V_1$ and $\gamma = \bar{\gamma}$, equation (3.7) is solved for k and the value of the design variable m_i . Then equations (3.9) and (3.8) are used to compute $\frac{\partial \gamma}{\partial V}$ as before. In general $\frac{\partial \gamma}{\partial V} \neq 0$ and a one-dimensional Regula Falsi process is initiated by repeating the process for another chosen value $V = V_2$.

Numerical evaluation of the approaches outlined could not be accomplished within the scope of this study.

3.5 Recommendation

The two-dimensional Regula Falsi procedure is recommended for inclusion in the Flutter Optimization Module for providing point solutions of the flutter equation. The procedure is more direct than any of the other procedures considered. It aims at roots of the flutter equations, either flutter speed and frequency or design variable and frequency, of which for every flight condition there are considerably fewer present than there are in-flight modes represented in the problem formulation. As a result, convergence on the wrong root would seem to be less likely than when modal solutions are sought. That the same procedure can be used for solving for different pairs of unknowns is considered an added advantage. In addition, it is equally applicable to the p-, the k- and the p-k method of formulating the flutter equation. A preliminary program is available that has shown good convergence behavior under a wide variety of input data.

Since it seems likely that the capability of directly solving for the point for which $\frac{\partial \gamma}{\partial V} = 0$ will be a factor in developing methods of flutter optimization, numerical test cases should be conducted to evaluate the methods related to determining the minimum damping in the hump mode. The results may influence the development of methods of optimization that take into account damping constraints.

4. MODALIZATION

4.1 General

Modalization is the reduction of the number of degrees of freedom by establishing modes of displacement in which the original degrees of freedom (usually point displacements) have a fixed relation to each other.

Let $\{z^{(i)}\}$ define a relation between the discrete structural displacements z . The arbitrary column matrix of displacements $\{z\}$ can then be approximated by linear combination of several linearly independent columns $\{z^{(i)}\}$:

$$\{z\} = \left[\{z^{(1)}\} \{z^{(2)}\} \dots \right] \begin{Bmatrix} q_1 \\ q_2 \\ \vdots \end{Bmatrix} \quad (4.1)$$

or in short notation:

$$\{z\} = [\bar{z}]\{q\} \quad (4.2)$$

The modalized flutter equation is:

$$[\bar{z}]^T \left[\frac{v^2}{c^2} [M] p^2 + (1+ig)[K] - 1/2 \rho v^2 [A(ik)] \right] [\bar{z}]\{q\} = 0 \quad (4.3)$$

Modalization is desirable whenever the total number of initial degrees of freedom is so large that solving the unmodalized equation becomes uneconomical, and is necessary if the number of initial degrees of freedom exceeds the capacity of the available computer program to solve the original characteristic value problem. Since, in general, the flutter equation is solved more frequently than the vibration equation and, in addition, the flutter equation must admit complex numbers, modalization is usually associated with the flutter equation. However, when using all the structural displacements of a detailed finite element structural model as degrees of freedom, modalization may be desirable or necessary for the vibration analysis as well.

In any discussion of modalization, the type of modes and the number of modes to be used must be considered. When used in an optimization procedure, the question of "updating" must be considered. Updating in this context means the adjustment of the modes after resizing the structural elements in the course of the optimization procedure. These three aspects of modalization will be discussed separately in the following sections.

4.2 Types of Modes

Before the advent of the high-speed computer, modalization (e.g., Rayleigh-Ritz method) was required even for vibration analyses. Relatively few and simple modes were used. With the increasing capacity of computers, the need for modalizing the vibration equation has all but disappeared. Thus, present practice is to determine natural vibration modes of the entire airplane from an unmodalized vibration equation and to use a certain number of modes, associated with the lower range of natural frequencies, to reduce the order of the flutter equation. For special investigations, such as the inclusion of actual control-surface-actuator impedances, or the entire automatic control system, additional control surface modes may be necessary.

In several instances in the literature (e.g., Reference 10), the use of component modes has been described. Component modes define the relations between discrete displacements of airplane components such as the wing or

fuselage, and are obtained by a vibration analysis in which only displacements of a particular component are used as degrees of freedom. Complications arise when the connections between components involve many structural displacements. Reference 11 shows, with a simple beam as an example, that the unjudicious use of component modes can give inaccurate results for even the lowest frequency of the entire body. The use of component modes is only recommended for the determination of natural vibration modes of the complete vehicle, and then only if it is necessary to reduce the order of the vibration equations.

Analytical modes, such as defined by polynomials and modes corresponding to static deflections, would obviate the need for repetitive vibration analyses during the optimization process if they are used as fixed modes. However, usually a considerably larger number of such modes is required, for the same accuracy of the flutter solution, than when vibration modes are used. No advantages off-setting that disadvantage have been encountered.

Specifically, analytical modes have been suggested for efficient generation of generalized aerodynamic force coefficients, as discussed in Section 5. The use of analytical modes may permit the analytical integration of the product of deflection and pressure modes. It makes it possible to compute invariant generalized aerodynamic force coefficients that can be combined linearly to form generalized aerodynamic forces for any arbitrary mode. To take advantage of this feature, however, the number of analytical modes must be large, since it must be adequate for a large number of stiffness and inertia distributions. The analytical modes thus can serve as reference modes that are the degrees of freedom for all vibration analyses from which a smaller number of vibration modes is obtained for use in flutter calculations. However, a large number of vibration modes of a basic configuration also can be used as reference modes and one would expect that fewer reference modes are needed if they are vibration modes than if they are analytical modes.

It was thought that using the (complex) flutter mode of a base configuration might reduce the number of modes required for an adequate flutter solution of a modified configuration. Some preliminary work during this study was done, but was not carried far enough for any conclusion to be drawn.

4.3 Number of Modes

The number of modes used in the flutter equation is of importance for the accuracy of the computed flutter speed and flutter speed derivatives with respect to design variables. At present there seems to be no readily available general criterion for determining the number of modes needed for a desired accuracy.

When trying to economize by restricting the number of modes to be used in flutter calculations, there is a need to frequently check whether the number of

modes is sufficient for accurate prediction of the flutter speed for arbitrary configurations. Thus there is an advantage in using flutter analysis procedures that allow a large number of modes even if that raises the cost of each individual flutter solution.

It has been pointed out in Reference 12, and it was confirmed by limited numerical analysis during this study, that more modes are needed for accurately computing flutter speed derivatives than for computing flutter speeds.

In deciding on the number of modes the computer environment may be an important factor to be judged by the analyst in addition to the accuracy required. Even the method of computer cost appropriation may influence the decision.

4.4 Updating of Modes

As resizing steps accumulate during the optimization process, the vibration modes of the initial configuration become less suited to accurately represent the revised structure. Ideally, therefore, after each resizing step a new vibration analysis should determine new modes for modalizing the flutter equation. The need for such updating is closely related to the number of modes used and the type and magnitude of structural changes incurred by the resizing. The use of a large number of modes tends to reduce the need for frequent updating. However, insufficient updating can cause the resizing steps to follow a zig-zag path that, in the extreme, may not converge.

The physical explanation for this is the following. Let the optimization procedure indicate a local stiffening as the optimum resizing for resizing step j . Then the vibration modes for step $j+1$ would show a decrease in local deformation. If the vibration modes for step j are used for step $j+1$, the excess local deformation tends to reinforce and overestimate the beneficial effect of that local stiffening. Thus, in the absence of modal updating, material tends to be added where the first resizing step, with the modes used, indicates where it is most beneficial.

Modal updating must not be confused with making the modal matrix a function of the design variables. This aspect of modalizing was recently introduced by Reference 12 and it is formulated in Reference 1. Determining each resizing step under the assumption of constant modes (i.e., independent of the design variables), but using updated modes at each resizing step, may underestimate the amount of material to be added locally for a certain amount of stiffening in a particular step, but it is not expected to cause an erratic or nonconvergent resizing path.

As important as the frequency of updating is on the efficiency of the optimization procedure, the number of modes used to do the final flutter analysis is more important from a general point of view since it provides the

final check on the optimization procedure. In the opinion of the present investigators, a check flutter analysis using a proven sufficient number of vibration modes of the final configuration should conclude any optimization process. If flutter requirements are not met, then a new optimization process can be initiated and, probably, more modes or more frequent updating, or both, should be used.

4.5 Recommendations

In view of experience during the present study, and as a result of experience with flutter analyses of actual airplane designs, the present investigators recommend the following:

1. A flutter module should provide the option of inputting arbitrary initial modalizing matrices or of generating initial modalizing matrices based on a vibration analysis of the initial configuration.
2. The number of vibration modes to be used for the flutter calculations should be an input option.
3. The frequency of updating the vibration modes should be an input option.
4. An option should be included to provide the analyst with information to determine whether his choice of number of modes and frequency of updating has led to satisfactory flutter characteristics. Such information might be provided by a vibration and flutter analysis of the final configuration with more modes than were used throughout the resizing process, a check on whether the optimality criteria for flutter are satisfied, or other check procedures.

5. AERODYNAMICS

5.1 Introduction

One of the objectives of this study is to develop general, efficient and accurate computational procedures for evaluating the unsteady aerodynamic parameters necessary for use in a flutter optimization module, without, however, evaluating aerodynamics theories.

The procedure should be general. That is, it should be applicable to all present, and hopefully future, theoretical formulations of unsteady aerodynamics.

The procedure should be efficient. In the context of application in a flutter optimization module, this implies a minimum of computational operations required to recompute the generalized aerodynamic force coefficients each time a modal updating occurs.

The procedure should be accurate. This implies it should be able to accommodate the most sophisticated formulations of the aerodynamics, such that the aerodynamics used in the flutter optimization module have the same accuracy as the aerodynamics used in a flutter analysis module.

In the following section general background for a matrix formulation that allows a procedure satisfying these requirements is presented. It is followed by the definition of the formulation and a discussion of how the dimensions of the matrices, the method of interpolation for modal deflections and arbitrary values of the reduced frequency k , and the number of reduced frequency intervals to be considered determine the sequence of operations that is most efficient. Conclusions and recommendations regarding the aerodynamics subroutine in a flutter optimization module are presented.

5.2 General

The elements of the matrix of generalized aerodynamic force coefficients are defined by:

$$A_{ij} = \iint f_i(x,y) p_j(x,y) dx dy \quad (5.1)$$

Here $p_j(x,y)$ is the lifting pressure distribution associated with an angle-of-attack distribution, $\alpha_j(x,y)$, which is defined by:

$$\alpha_j(x,y) = \left\{ \frac{ik}{c} f_j(x,y) + \frac{\partial f_j(x,y)}{\partial x} \right\} q_j \quad (5.2)$$

which expresses α_j as the sum of $\frac{\dot{z}}{V}$ and $\frac{\partial z}{\partial x}$ terms in the case of harmonic motion with reduced frequency k in a mode defined by $f_j(x,y)$.

Expressed in the form of equation (5.1), the evaluation of A_{ij} requires evaluation of the surface integral each time new modes f_i are used. In the usual flutter investigation many different sets of modes are used, corresponding to different weight and stiffness distributions. In addition it is expected that frequent remodalization is required in an optimization procedure.

Thus it is advantageous to develop a method in which the generalized aerodynamic force coefficients are formed from a mode-independent part that contains as many of the numerical operations as possible, and a relatively simple mode-dependent part.

Four different approaches are recognized in separating mode-independent operations from mode-dependent operations. One method relies entirely on analytical evaluation of the surface integral (Equation (5.1)). A second method formulates a numerical evaluation of the surface integral leading, effectively, to "lumped" aerodynamic forces at a grid of integration points. In a third method pressure distribution modes are analytically integrated over small areas and combined into elementary aerodynamic forces directly comparable to, and treated as, inertial forces. The fourth method recognized is based on a finite element approach, the basic formulation of which has no reference to pressure distributions over the entire surface.

The first three methods are usually thought of as stemming from the kernel function approach of Reference 13. In it the pressure distribution

$p_j(x,y)$ is assumed to be a linear combination of pressure distribution modes $p^n(x,y)$:

$$p_j(x,y) = \sum a_j^n p^n(x,y) \quad (5.3)$$

The pressure mode coefficients a_j^n are determined from a boundary condition requiring that the normalized induced velocity distribution resulting from the pressure distribution equals the angle-of-attack distribution at a set of downwash collocation points:

$$\{a_j^n\} = [PKI]^{-1} \{\alpha_j\} \quad (5.4)$$

Combining equations (5.3) and (5.4) leads to

$$\{p_j\} = [p^n] [PKI]^{-1} \{\alpha_j\} \quad (5.5)$$

where the elements of matrix $[PKI]$ are the integrals of the product of pressure distribution mode and an aerodynamic kernel.

The columns of $[p^n]$ are linearly independent pressure distribution modes.

5.2.1 Analytical Integration - When $p_j(x,y)$ is a linear combination of pressure distribution modes $p^n(x,y)$, analytical modal functions $f_k(x,y)$ can be selected such that the integrals $\iint f_k(x,y) p^n(x,y) dx dy$ can be evaluated analytically. Generalized aerodynamic force coefficients in terms of modal coordinates can then be formed. The analytical modes can be used as arbitrary modes to modalize the flutter equation, or a mode-dependent transformation between the analytical modes and the actual modes is used to express the generalized aerodynamic force coefficients in terms of the actual modal coordinates.

5.2.2 Numerical Integration of the Product of Displacement and Pressure - Reference 12 defines an approach to separating mode-independent operations from mode-dependent operations in which the surface integral of equation (5.1) is evaluated numerically. A Gaussian integration procedure is suggested to evaluate the integral. The pressure $p_j(x,y)$ and the deflection $h_i(x,y)$ are evaluated at integration points defined by the Gaussian procedure. Weighting factors in the form of a row matrix, $[WF]$, make it possible to write:

$$\iint h_i(x,y) p_j(x,y) dx dy \approx [WF] \{h_i p_j\} \quad (5.6)$$

The right hand side of equation (5.6) can be written as:

$$[WF] \{h_i p_j\} = [WF] [h_i] \{p_j\} = [h_i] [WF] \{p_j\} \quad (5.7)$$

The interchange of the row and diagonal matrix in the latter part of the combined equation (5.7) makes it possible to separate the mode-dependent operations from the mode-independent operations.

The column matrix $\{WF \cdot p_j\}$ is a set of lumped aerodynamic forces. The deflections h_i can be expressed in terms of the deflections z_i at the structural nodes by the relation.

$$\{h_i(x,y)\} = [H] \{z_i\} \quad (5.8)$$

A variation of this method is obtained if instead of the pressure distribution the velocity potential distribution $\phi_j(x,y)$ due to an

angle-of-attack distribution $\alpha_i(x,y)$ is used. With the familiar linearized relation between pressure and velocity potential:

$$p = -2\left(\frac{\partial\varphi}{\partial x} + ik\varphi\right) \quad (5.9)$$

equation (5.1) becomes:

$$A_{ij} = -2ik \iint f_i \varphi_j dx dy - 2 \iint f_i \frac{\partial\varphi_j}{\partial x} dx dy \quad (5.10)$$

It can be shown that with the help of numerical techniques equation (5.10) can be written as:

$$A_{ij} \approx \begin{bmatrix} h_i \end{bmatrix} \left[ik \begin{bmatrix} WF \end{bmatrix} + \begin{bmatrix} WFD \end{bmatrix} \right] \begin{Bmatrix} \varphi_j \end{Bmatrix} \quad (5.11)$$

where $[WF]$ performs numerically the first integration in equation (5.10) and $[WFD]$ performs the differentiation and integration in the second term of that equation. Equation (5.11) is a triple matrix product, similar to equation (5.7), in which the center matrix is mode independent. For additional details see Reference 1.

5.2.3 Numerical Integration of the Pressures - When $p_j(x,y)$ is a linear combination of pressure distribution modes $p^n(x,y)$, the integral

$\iint p^n(x,y) dx dy$ can be evaluated over small areas, often referred to as

aerodynamic boxes, into which the surface is divided. By evaluating

$\iint x p^n(x,y) dx dy$ and $\iint y p^n(x,y) dx dy$ over the same areas, lumped

aerodynamic forces can be determined in magnitude and location. The modal displacement at the location of each lumped force (i.e., for each aerodynamic

box and each $p^n(x,y)$) can be expressed in terms of the structural degrees

of freedom. Thus the product of each lumped force and its modal displacement can be formed. Summation over the aerodynamic boxes and the pressure dis-

tribution modes participating in $p_j(x,y)$ leads to A_{ij} .

5.2.4 Finite Element Approach - In a finite element approach, lumped aerodynamic forces corresponding to $\{WF \cdot p_j\}$ (see equation (5.7)) are expressed

directly in terms of angle-of-attack distributions $\{\alpha_j\}$ under appropriate simplifying assumptions. The generalized aerodynamic force coefficients are formed as in Section 5.2.2.

5.3 Basic Formulation

Whatever the approach, or whatever aerodynamic theory is chosen, the generalized aerodynamic force coefficients in terms of modal coordinates can be expressed as the product of five matrices of which only the first and last are mode-dependent:

$$[A_{ij}] = [\bar{z}]^T [H]^T [AIC] [W] [\bar{z}] \quad (5.12)$$

The matrix $[AIC] = [AIC(k)]$, a function of the reduced frequency $k = \frac{\omega c}{V}$ and the Mach number, is the core of the aerodynamics and is independent of mode shape. Its elements are basic aerodynamic influence coefficients defining lumped aerodynamic forces $\{Z_a\}$ at an aerodynamic force grid in terms of the angles of attack at downwash collocation points:

$$\{Z_a\} = [AIC(k)] \{\alpha\} \quad (5.13)$$

$[W] = [W(k)] = [DX] + ik[DZ]$ relates the angles of attack to the structural displacements $\{z\}$. It is independent of mode shape.

The matrix $[H]^T$ is independent of k and of mode shape, and distributes lumped aerodynamic forces and moments over the structural coordinates.

In the case that the approach of Section 5.2.1 is followed, $[AIC]$ is the matrix of generalized aerodynamic force coefficients in terms of the analytical modes; $[H]$ and $[W]$ are equated to

$$[F] = \left[[f_\ell(x,y)]^T [f_\ell(x,y)] \right]^{-1} [f_\ell(x,y)]^T \quad (5.14)$$

where columns of $[f_\ell(x,y)]$ are the fixed analytical modes. The matrix $[F]$ is independent of k and of the actual mode shapes used to reduce the order of the flutter equation.

The operations performed by $[W]$ and $[H]^T$ may be included in $[AIC]$. Equation (5.12) then reduces to the product of three matrices.

The matrix $[\bar{z}]$ contains the modal columns in terms of the structural deflections $\{z\}$.

The matrices $[AIC]$, $[H]^T$ and $[W]$ are constant during an optimization procedure. They will be used many times during the design process of an airplane with a given external configuration. It is therefore advantageous to form these matrices in a special aerodynamics computer program.

Each time during an optimization procedure that a remodalization takes place, $[A_{ij}]$ must be recomputed. Depending on the dimensions of the matrices in equation (5.12), it may be more efficient to compute the triple matrix product

$$[HAW] = [H]^T [AIC] [W] \quad (5.15)$$

in the aerodynamics program, or to perform one or both of the multiplications $[\bar{z}]^T [H]^T$ and $[W][\bar{z}]$ in the optimization program.

In the following sections factors are discussed that must be considered in determining which approach to numerically evaluating $[A_{ij}]$ according to equation (5.12) is most efficient.

5.4 Factors Affecting The Efficiency of The Numerical Evaluation of The Matrix of Generalized Aerodynamic Force Coefficients

It is believed that the formulation

$$[A_{ij}] = [\bar{z}]^T [HAW] [\bar{z}] \quad (5.16)$$

which follows from combining equations (5.12) and (5.15), is widely used in industry. Detailed study of the formulation of equation (5.12), which directly follows from Reference 12, however, indicates that there are conditions under which it is more efficient to compute $[\bar{z}]^T[H]^T$ and/or $[W][\bar{z}]$

in the flutter optimization module. Extensive comparisons have been made and are discussed in detail in Reference 1. In the following the factors affecting these comparisons are discussed and major conclusions are presented.

5.4.1 Matrix Population - When matrices are sparsely populated, or populated in well defined blocks, proper programming can take advantage of this.

In equation (5.12) the matrices $[W]$ and $[H]$ may be sparsely populated. These matrices perform an interpolation and $[W]$, in addition, determines streamwise slopes at collocation points. In the case of simple interpolation (linear or low degree polynomial), each row in $[W]$ expresses the angle of attack at a downwash collocation point in terms of several surrounding structural coordinates. Similarly, each row of $[H]$ expresses the deflection at an integration point in terms of several surrounding structural coordinates. Thus each row in $[W]$ and $[H]$ contains relatively few, say < 20 , nonzero elements; for linear interpolation each row contains four nonzero elements. In the case of interpolation by the surface spline method, the matrices $[W]$ and $[H]$ are fully populated, at least in the blocks that cover the aerodynamic surfaces.

Without specific stipulations, equation (5.16) implies that the order of the matrix $[HAW]$ equals the number of structural coordinates. There may be a considerable number of structural coordinates that do not carry an aerodynamic load. They correspond to zero elements in $[HAW]$. It is not expected that the fraction of nonzero elements will be high enough to justify treating $[HAW]$ as a sparse matrix. However, by proper ordering of the structural coordinates, the nonzero elements in $[HAW]$ may be concentrated in one or more blocks. Then the aerodynamics program may form $[HAW]$ based on aerodynamic load carrying structural coordinates only. Correspondingly, the flutter optimization module must eliminate structural coordinates that carry no aerodynamic load from the modalizing matrix $[\bar{z}]$.

5.4.2 Interpolation for Arbitrary k Value - It is generally accepted that when the generalized aerodynamic force coefficients are determined for a discrete set of values, k_ℓ , of the reduced frequency, $[A_{ij}(k_\ell)]$ interpolation is adequate for approximating $[A_{ij}(k)]$ at arbitrary values of k .

Two methods of interpolation are considered: cubic polynomial and cubic spline.

can lead to "hunting" (oscillation between k values in adjacent intervals).

Therefore it is recommended to define $[A'_{ij}(k)]$ by:

$$[A'_{ij}(k)] = \sum_{\ell=1}^4 \mathcal{L}_{\ell}(k) [A'_{ij}(k_{\ell})] \quad (5.20)$$

where $[A'_{ij}(k_{\ell})]$ is the derivative of $[A_{ij}(k)]$ evaluated at $k=k_{\ell}$. $[A'_{ij}(k_{\ell})]$ is an input to the interpolation subroutine. Thus the difference between equations (5.19) and (5.20) is that in equation (5.19) differentiation occurs after the polynomial fit and in equation (5.20) it occurs before the polynomial fit. The formation of $[A_{ij}(k_{\ell})]$ in the flutter optimization module is based on equation (5.12), equation (5.16) or any variant that is chosen as being most appropriate. The derivative $[AIC'(k)]$ or $[HAW'(k)]$ is needed and should be calculated outside the flutter optimization module by any method that gives adequate accuracy.

To define $[A_{ij}(k)]$ and $[A'_{ij}(k)]$ in one k interval, four matrices $[AIC(k_{\ell})]$ and four matrices $[AIC'(k_{\ell})]$, or four of each of the matrices $[HAW(k)]$ and $[HAW'(k)]$ must be input into the flutter optimization module. If $[AIC(k)]$ and $[AIC'(k)]$ are input, $[A_{ij}(k)]$ follows from equation (5.12). $[A'_{ij}(k)]$ is given by:

$$[A'_{ij}(k)] = [\bar{z}]^T [H]^T [AIC'(k)] [W] [\bar{z}] + i [\bar{z}]^T [H]^T [AIC(k)] [DZ] [\bar{z}] \quad (5.21)$$

If k moves to an adjacent interval only two of the input matrices, one for the aerodynamics coefficients and one for their derivatives, need be replaced.

It should be noted that, in effect, each k interval has its own associated polynomials for the value of $[A_{ij}(k)]$ and its derivative.

The cubic spline method also defines different cubic polynomials for each k interval. The coefficients for the polynomial, however, are derived

from matrices defined for all k values k_ℓ , $\ell = 1, 2 \dots n$. They follow from the assumption of continuity of derivatives over the complete range of k values. The resultant expression, e.g., for $[AIC(k)]$, is:

$$[AIC_{ij}^\ell(k)] = [AIC_{\ell 0}] + [AIC_{\ell 1}](k-k_\ell) + [AIC_{\ell 2}](k-k_\ell)^2 + [AIC_{\ell 3}](k-k_\ell)^3 \quad (5.22)$$

The matrices $[AIC_{\ell 0}]$ to $[AIC_{\ell 3}]$ should be formed outside the flutter optimization module.

Then:

$$[A_{ij}^\ell(k)] = [AX_{\ell 0}] + [AX_{\ell 1}](k-k_\ell) + [AX_{\ell 2}](k-k_\ell)^2 + [AX_{\ell 3}](k-k_\ell)^3 + [AZ_{\ell 0}]ik + [AZ_{\ell 1}]ik(k-k_\ell) + [AZ_{\ell 2}]ik(k-k_\ell)^2 + [AZ_{\ell 3}]ik(k-k_\ell)^3 \quad (5.23)$$

where

$$[AX_{\ell 0}] = [\bar{z}]^T [H]^T [AIC_{\ell 0}] [DX] [\bar{z}] \quad \text{etc.}, \quad (5.24)$$

and

$$[AZ_{\ell 0}] = [\bar{z}]^T [H]^T [AIC_{\ell 0}] [DZ] [\bar{z}] \quad \text{etc.}, \quad (5.25)$$

Because of the implied continuity of the derivatives, it is proper to differentiate equation (5.23) directly and thus no additional matrices for the derivative need to be formed.

To define $[A_{ij}(k)]$ and $[A'_{ij}(k)]$ in one k interval if the aerodynamics input is $[AIC_{\ell 0}]$ to $[AIC_{\ell 3}]$ requires eight coefficient matrices.

Switching k to any other interval requires replacing all eight matrices.

If the basic aerodynamics input is in the form of $[HAW_{\ell 0}]$ to $[HAW_{\ell 3}]$, then only four coefficient matrices are needed for each k interval.

5.4.3 Number of k Intervals - Let the basic aerodynamics input into the flutter optimization module be $[HAW(k_\ell)]$; the number of k intervals to be considered is ℓ .

For the cubic polynomial, Lagrange's interpolation formula is considered to be most efficient since it expresses $[A_{ij}(k)]$ directly in terms of its value $[A_{ij}(k_\ell)]$ at discrete values k_ℓ , $\ell = 1, 2, 3, 4$:

$$[A_{ij}(k)] = \sum_{\ell=1}^4 \mathcal{L}_\ell(k) [A_{ij}(k_\ell)] \quad (5.17)$$

where $\mathcal{L}_1(k)$ is defined by:

$$\mathcal{L}_1(k) = \frac{(k-k_2)(k-k_3)(k-k_4)}{(k_1-k_2)(k_1-k_3)(k_1-k_4)} \quad (5.18)$$

and cyclic substitution leads to \mathcal{L}_2 , \mathcal{L}_3 and \mathcal{L}_4 .

The interpolation formula (5.17) is used only for the interval $k_2 < k < k_3$. For the interval $k_3 < k < k_4$ the index ℓ must be increased by one and for $k_1 < k < k_2$ the index must be lowered by one.

Since most methods of optimization require the computation of the derivative of the aerodynamics matrix with respect to k , the formation of the derivative, $[A'_{ij}(k)]$, must be considered.

Differentiating equation (5.17) with respect to k leads to an expression:

$$[A'_{ij}(k)] = \sum_{\ell=1}^4 \mathcal{L}'_\ell(k) [A_{ij}(k_\ell)] \quad (5.19)$$

that is based on the same aerodynamic matrices as equation (5.17). This approach, however, combined with the re-indexing of k_ℓ as k moves to an adjacent interval, leads to jumps in the value of $[A'_{ij}(k)]$ at all values $k=k_\ell$. Apart from considerations of accuracy, this is undesirable since it

If cubic polynomial interpolation is used, $(\ell+3)$ matrices $[HAW(k_\ell)]$ and $[HAW'(k_\ell)]$ must be input and pre- and postmultiplied by $[\bar{z}]^T$ and $[\bar{z}]$ to form the $2(\ell+3)$ matrices $[A_{ij}(k_\ell)]$ and $[A'_{ij}(k_\ell)]$ needed in Lagrange's interpolation formula.

If cubic spline interpolation is used, 4ℓ matrices $[HAW_{\ell 0}]$ to $[HAW_{\ell 3}]$ must be input and pre- and postmultiplied by $[\bar{z}]^T$ and $[\bar{z}]$ to form 4ℓ coefficient matrices $[A_{\ell 0}]$ to $[A_{\ell 3}]$.

Under otherwise equal circumstances, polynomial interpolation is more efficient if

$$2(\ell+3) < 4\ell \quad \text{or} \quad \ell > 3 \quad (5.26)$$

This condition is valid for other sequences of operations to form $[A_{ij}(k)]$ according to equation (5.12). However, there are also sequences for which $\ell \geq 4$ or $\ell \geq 5$ is required for cubic polynomial interpolation to be more efficient than cubic spline interpolation.

The number of intervals that should be used is difficult to predict. If only one flutter constraint is active, k may stay within a rather small range during the entire optimization process and that range may lie completely within one k interval. Obviously only aerodynamic matrices applicable to that one k interval need be computed. In general, however, several k intervals are required.

5.4.4 Sequence of Multiplications - Defining one computational operation as one multiplication and one addition, the numbers of such operations required inside the flutter optimization module for different sequences of multiplications in equation (5.12) have been determined and compared.

The following options have been considered; the numerals indicate the sequence of multiplication.

$$H1: \quad [A_{ij}] = \underbrace{\underbrace{[\bar{z}]^T \cdot [H]^T}_{1} \cdot [AIC]}_3 \cdot \underbrace{[W] \cdot [\bar{z}]}_2 \quad (5.27)$$

$$\underbrace{\hspace{10em}}_4$$

$$H2: \quad [A_{ij}] = \underbrace{\underbrace{[\bar{z}]^T \cdot [H]^T}_{1} \cdot [AIC]}_3 \cdot \underbrace{[W] \cdot [\bar{z}]}_2 \quad (5.28)$$

$$\underbrace{\hspace{10em}}_4$$

H3: $[AIC] \cdot [W] = [AW]$ is computed outside the flutter optimization module

$$[A_{ij}] = \underbrace{\underbrace{[\bar{z}]^T \cdot [H]^T}_{1} \cdot [AW]}_3 \cdot [\bar{z}] \quad (5.29)$$

H4: $[H]^T [AIC] = [HA]$ is computed outside the flutter module

$$[A_{ij}] = \underbrace{\underbrace{[\bar{z}]^T \cdot [HA]}_1 \cdot [W]}_3 \cdot [\bar{z}] \quad (5.30)$$

H5: $[H]^T [AIC] [W] = [HAW]$ is computed outside the flutter module

$$[A_{ij}] = [\bar{z}]^T [HAW] [\bar{z}] \quad (5.31)$$

The number of computational operations is independent of sequence of multiplications in H5.

Formulas defining the number of numerical operations have been derived and are reported in Reference 1. No option stands out as clearly superior or inferior to all others, but some comments are offered in Section 5.5.

5.4.5 Form of Inputting the Angle-of-Attack Generating Matrix - In the previous section the options are defined as if one complex matrix $[W(k)]$ was input for each value of k . Suboptions of the options that require inputting $[W]$ can be obtained by considering the definition of $[W]$.

$$[W(k)] = [DX] + ik[DZ] \quad (5.32)$$

Thus in options H1, H2 and H4 an "a" and a "b" version can be recognized. In the "a" version $[W(k)]$ is input for several k values. In the "b" version the real matrices $[DX]$ and $[DZ]$ are input.

Whether option "a" is more efficient than option "b" depends on the other factors. In the case of option H1 it seems that H1a is favored if interpolation for deflections is held simple. Option H1b is favored if surface spline interpolation is used. If the "a" option is used and the derivative of the aerodynamics matrix is needed, matrix $[DZ]$ must be input anyway.

5.5 Summary of Comparisons

Detailed comparison of the options H1 through H5 is reported in Reference 1. The following summarizes the comparisons.

5.5.1 Input Storage Requirements - If the number of k intervals to be used is three or more, cubic polynomial interpolation for arbitrary values of k requires less input storage than cubic spline interpolation for all options H1 through H5.

5.5.2 Core Space - For options H5 and H3, cubic polynomial interpolation for k requires two times as much core space as cubic spline interpolation. For all other options, both methods of interpolation require the same core space.

5.5.3 Read-In - Cubic polynomial interpolation for arbitrary k requires less read-in than cubic spline interpolation as the value of k moves into an adjacent interval.

5.5.4 Number of Computational Operations - Cubic polynomial interpolation for arbitrary values of k requires fewer computational operations than cubic spline interpolation under the following conditions:

options H3 and H5: if the number of k intervals is more than three.

options H1 and H4: if the number of k intervals is more than four.

option H2: if the number of k intervals is more than five.

Which of the options H1 through H5 is most efficient depends strongly on the dimensions of the matrices. These, in turn, depend on the desired accuracy, and the methods used for integrating or lumping the aerodynamic pressures and interpolating and differentiating the structural displacements.

Let M be the number of modes, N the number of structural coordinates before modalizing, D the number of downwash collocation points and K the number of integration points. Then equation (5.12) can be annotated as follows:

$$\begin{bmatrix} A_{ij} \end{bmatrix} = \begin{bmatrix} \bar{z} \end{bmatrix}_{M,N}^T \begin{bmatrix} H \end{bmatrix}_{N,K}^T \begin{bmatrix} AIC \end{bmatrix}_{K,D} \begin{bmatrix} W \end{bmatrix}_{D,N} \begin{bmatrix} \bar{z} \end{bmatrix}_{N,M} \quad (5.33)$$

From this equation it can be seen that if K and D are small compared with N it becomes advantageous to perform the multiplications $\begin{bmatrix} \bar{z} \end{bmatrix}^T \cdot \begin{bmatrix} H \end{bmatrix}^T$ and $\begin{bmatrix} W \end{bmatrix} \cdot \begin{bmatrix} \bar{z} \end{bmatrix}$ in the flutter optimization module. If K and D are equal to N or larger, then it becomes advantageous to form the product

$\begin{bmatrix} H \end{bmatrix}^T \begin{bmatrix} AIC \end{bmatrix} \begin{bmatrix} W \end{bmatrix}$ outside the flutter module. The relationships defining

when an option is better than another are complicated. They are documented in detail in Reference 1, but they have not led to simple criteria.

5.6 Conclusions and Recommendations

The preceding sections lead to the following conclusions:

1. Formulation of the generalized aerodynamic forces in the form

$$\begin{bmatrix} A_{ij}(k) \end{bmatrix} = \begin{bmatrix} \bar{z} \end{bmatrix}^T \begin{bmatrix} H \end{bmatrix}^T \begin{bmatrix} AIC(k) \end{bmatrix} \begin{bmatrix} W(k) \end{bmatrix} \begin{bmatrix} \bar{z} \end{bmatrix}$$

is possible and practicable for all approaches to determining unsteady aerodynamic forces.

2. Several options of inputting the matrices $[H]^T$, $[AIC(k)]$, and $[W(k)] = [DX] + ik[DX]$ can be recognized, i.e., separately, after multiplication, in the form of cubic spline matrices, or in derivative form. Which option is most efficient depends strongly on sizes of the matrices, whether the derivative of the aerodynamics matrix is needed, the method of interpolation for arbitrary k , the population of the matrices $[H]$ and $[W]$ and the number of k intervals expected to be active during the optimization process.
3. In addition to depending on the number of computational operations required to form $[A_{ij}(k)]$ the efficiency of the flutter optimization process may also depend on the required input storage, the possibility of storing all matrices required for interpolation of the aerodynamics for k in one interval in core, and the read-in required if the value of k moves to an adjacent interval.
4. It is possible to design a flutter optimization module that includes all options such that the user can choose the option that is most efficient, that fits his available data or that he prefers for some other reason.

In view of the above conclusions the following recommendation is made:

In designing a flutter optimization module for a facility, the calculation of the generalized aerodynamic force coefficients and their derivatives should be based on the formulations presented in this section. That is, a mode-independent part should be generated outside the flutter optimization module leaving an often to be repeated mode-dependent part of the calculations to be performed inside the flutter module. The number of options is large and it may not be practicable to include all options in the flutter module. The choice of options is facility dependent. Certain practices of generating generalized aerodynamic force coefficients may already exist and the existing computer system may influence the choice significantly. The module, however, should allow the user considerable freedom in choosing the option that is most efficient for his problem. The number of options to be included should be decided on the basis of a stand-alone flutter optimization module. It should not be restricted because of the module being part of a general analysis system which at present has only a restricted choice of outputting aerodynamic coefficients.

6. METHODS OF OPTIMIZATION FOR FLUTTER

6.1 General

In this section five approaches to structural optimization with flutter constraint are reviewed. All five belong to the category of direct methods; i.e. methods in which the mathematical formulation defines resizing steps aimed directly at determining the extreme value of the objective function, in this case minimizing the structural mass. In contrast, in the indirect methods the mathematical formulation defines resizing steps aimed at satisfying a criterion that, when satisfied, indicates that the optimum condition is reached.

The five methods reviewed represent distinctly different mathematical formulations. They are, in the order of review: the gradient methods of Rudisill and Bhatia (Reference 14), the weight gradient method of Simodynes (Reference 15), a penalty function method (Reference 16), a method of feasible directions (Reference 17), and a method that evolved from the method of Incremented Flutter Analysis (References 4 and 18). All are formulated under the assumption of modalization matrices that are independent of the design variables, but can be updated at any resizing step.

In addition to the qualitative evaluation of the methods presented in this section there is a numerical evaluation and comparison in Appendix A. In that appendix the results are presented of applying these methods to an optimization task in which the flutter equation is written in terms of 49 discrete degrees of freedom and is not modalized.

6.2 Rudisill-Bhatia Approach

Reference 14 defines four different resizing columns that alone or in combination can be used to design a minimum weight, or near minimum weight, structure that satisfies a flutter speed constraint. The resizing columns

are defined in terms of incremental values, ΔP_i , of the design variables

P_i . (The notation of Reference 14 is followed.) Three of them involve the

gradient of the flutter speed with respect to the design variables. In connection with this, Reference 14 presents closed form analytical expressions for the derivative of the flutter speed with respect to a design variable. These expressions have been used successfully in numerical test cases performed during this study. Numerical values of the derivatives are in good agreement with values obtained using the approach of Reference 15. In the following the resizing columns defined in Reference 14 are discussed. The terminology of Reference 14 is followed.

6.2.1 Velocity Gradient Search - Equation (23) of Reference 14 defines a column of design variable increments as:

$$\left\{ \Delta P_i \right\} = \frac{\Delta V}{\sum_{i=1}^n \left(\frac{\partial V}{\partial P_i} \right)^2} \left\{ \frac{\partial V}{\partial P_i} \right\} \quad (6.1)$$

Equation (6.1) defines a direction by means of $\left\{ \frac{\partial V}{\partial P_i} \right\}$. The magnitude of the increments is directly related to the velocity increment ΔV through the coefficient $\frac{\Delta V}{\sum_{i=1}^n \left(\frac{\partial V}{\partial P_i} \right)^2}$, where $\Delta V = \left[\frac{\partial V}{\partial P_i} \right] \left\{ \Delta P_i \right\}$ is the approximate increase in flutter speed due to the design variable increments $\left\{ \Delta P_i \right\}$.

The formulation is known as the method of steepest ascent of the function $V(P_i)$. The direction implied by equation (6.1) defines a column of directional cosines $\left\{ \frac{dP_i}{dS} \right\}$ for which $\left\{ \frac{dV}{dS} \right\} = \left\{ \frac{dV}{\sqrt{\sum (dP_i)^2}} \right\}$ is maximum.

Assuming a linear relationship between the design variables P_i and the associated mass, $m_i = C_i P_i$, equation (6.1) can be written as:

$$\left\{ \Delta P_i \right\} = \frac{\Delta V}{\sum \left(C_i \frac{\partial V}{\partial m_i} \right)^2} \left\{ C_i \frac{\partial V}{\partial m_i} \right\} \quad (6.2)$$

Taking $\frac{\partial V}{\partial m_i}$ as a reference it can be seen that the direction of $\left\{ \Delta P_i \right\}$ depends on the scaling between the design variable and the associated mass. Since the quantity of direct interest is total mass, and not design variables related to mass, it seems logical to choose elementary mass as design variables and choose $C_i=1$. In that case equation (6.1) becomes:

$$\left\{ \Delta m_i \right\} = \frac{\Delta V}{\sum_{i=1}^n \left(\frac{\partial V}{\partial m_i} \right)^2} \left\{ \frac{\partial V}{\partial m_i} \right\} \quad (6.3)$$

which represents a design change in the direction of maximum $\frac{dV}{\sqrt{\sum (dm_i)^2}}$.

The usefulness of the velocity gradient search is related to its ability to raise the flutter speed more efficiently, i.e., with a smaller increase in total mass m_i than a simple increase of the overall stiffness level. Simple physical considerations, however, lead to the conclusion that the most efficient move is in the direction in which $\frac{dV}{dM} = \frac{dV}{\sum dm_i}$ is maximum.

The numerical evaluations in Appendix A indicate that a resizing column proportional to $\left\{ \frac{\partial V}{\partial m_i} \right\}$ is an efficient means of resizing a structure in one step to satisfy a flutter speed constraint with a moderate mass penalty, thus providing a good starting point for a procedure that minimizes the total mass at constant flutter speed.

The relative efficiency of a resizing column proportional to $\left\{ \frac{\partial V}{\partial m_i} \right\}$ follows from the fact that it tends to add more material where it is most efficient in raising the flutter speed. Design variables for which $\frac{\partial V}{\partial m_i}$ is negative are reduced in value, which also raises the flutter speed.

6.2.2 Mass Gradient Search - Equation (30) of Reference 14 defines a column of design variable increments:

$$\left\{ \Delta P_i \right\} = \frac{\Delta V}{\left[\frac{\partial V}{\partial P_i} \right]} \left\{ \frac{\partial M}{\partial P_i} \right\} \quad (6.4)$$

where M is the total mass: $M = \sum m_i$.

Equation (6.4) defines a direction by means of $\left\{ \frac{\partial M}{\partial P_i} \right\}$. The magnitude of the increments is directly related to the velocity increment ΔV through the coefficient $\frac{\Delta V}{\left[\frac{\partial V}{\partial P_i} \right] \left\{ \frac{\partial M}{\partial P_i} \right\}}$.

The direction defined by equation (6.4) corresponds to a maximum value of

$$\sqrt{\frac{dM}{\sum (dP_i)^2}}$$

Again assuming a linear relation between the design variables P_i and the associated mass $m_i = C_i P_i$ it can be seen that

$$M = \sum_{i=1}^n m_i = \sum_{i=1}^n C_i P_i \quad (6.5)$$

and $\frac{\partial M}{\partial P_i} = C_i$.

Thus equation (6.4) becomes:

$$\{\Delta P_i\} = \frac{\Delta V}{\left[C_i \frac{\partial V}{\partial m_i} \right] \{C_i\}} \{C_i\} = \frac{\Delta V}{\sum_{i=1}^n C_i^2 \frac{\partial V}{\partial m_i}} \{C_i\} \quad (6.6)$$

It is apparent that the direction of $\{\Delta P_i\}$ again depends on the scaling between design variable and associated mass.

Choosing m_i as design variables equation (6.4) becomes:

$$\{\Delta m_i\} = \frac{\Delta V}{\sum_{i=1}^n \frac{\partial V}{\partial m_i}} \{1\} \quad (6.7)$$

which represents a uniformly distributed weight increment.

Reference 14 uses equation (6.4) to define a decrease in total mass, thus a negative ΔV is used. By following a path of steepest descent for $M(P_i)$ the emphasis is on decreasing total mass. The relation ΔM vs. ΔV is not considered.

The mass gradient search could be used in combination with the velocity gradient search to formulate an optimization procedure. Alternate application of these searches tends to lower the design variable weight required for satisfying the flutter constraint since the velocity gradient search tends to increase the flutter speed by adding a relatively efficient mass distribution. Although the mass gradient search removes mass indiscriminately, repeated application of both searches tends to an optimum mass distribution. Rather than remove mass indiscriminately, it seems logical to remove mass first where

it reduces flutter speed the least. That would be the case if most mass would be removed where $\frac{\partial V}{\partial m_i}$ is smallest. Note that if there are design variables for which $\frac{\partial V}{\partial m_i}$ is negative their reduction increases the flutter speed, allowing more mass to be removed from design variables with a small positive value of $\frac{\partial V}{\partial m_i}$. An efficient method to remove mass is to remove it proportionally to $\frac{1}{\partial V / \partial m_i}$ for design variables for which $\frac{\partial V}{\partial m_i}$ is positive.

In Reference 1 the sequential application of a velocity gradient search and a mass gradient search is replaced by the application of an equivalent two-component column of design variable increments. Regrouping of the elements in the two components shows clearly that this method does result in a resizing where more mass is added where $\frac{\partial V}{\partial m_i}$ is larger and more mass is removed where $\frac{\partial V}{\partial m_i}$ is smaller, regardless of the algebraic sign of $\frac{\partial V}{\partial m_i}$.

6.2.3 Gradient Projection Searches - The gradient projection search follows a direction of steepest ascent while satisfying a constraint. The gradient projection search used in Reference 14 is aimed at following the steepest ascent of the flutter speed as a function of design variables while keeping the total weight constant.

From equations (32) and (34) in Reference 14 the following column of design variable increments can be derived:

$$\begin{Bmatrix} \Delta P_i \end{Bmatrix} = \frac{\Delta S}{\sqrt{\sum \left[\left(\frac{\partial V}{\partial P_i} \right)^2 + \lambda_1 \frac{\partial V}{\partial P_i} \frac{\partial M}{\partial P_i} \right]}} \begin{Bmatrix} \frac{\partial V}{\partial P_i} + \lambda_1 \frac{\partial M}{\partial P_i} \end{Bmatrix} \quad (6.8)$$

where $\Delta S^2 = \sum_{i=1}^n (\Delta P_i)^2$, the step size in terms of design variables.

Equation (6.8) can also be written as

$$\begin{Bmatrix} \Delta P_i \end{Bmatrix} = \frac{\Delta V}{\left[\frac{\partial V}{\partial P_i} \right] \left[\frac{\partial V}{\partial P_i} + \lambda_1 \frac{\partial M}{\partial P_i} \right]} \begin{Bmatrix} \frac{\partial V}{\partial P_i} + \lambda_1 \frac{\partial M}{\partial P_i} \end{Bmatrix} \quad (6.9)$$

In both equations

$$\lambda_1 = - \frac{\left[\frac{\partial V}{\partial P_i} \right] \left\{ \frac{\partial M}{\partial P_i} \right\}}{\left[\frac{\partial M}{\partial P_i} \right] \left\{ \frac{\partial M}{\partial P_i} \right\}} \quad (6.10)$$

which follows from the condition

$$\Delta M = \left[\Delta P_i \right] \left\{ \frac{\partial M}{\partial P_i} \right\} = 0 \quad (6.11)$$

Usually $V(P_i)$ has a maximum for a given total weight. Although in the application shown in Reference 14 this maximum is not sought, it may be reached. It would seem, therefore, that the formulation of equation (6.8) with ΔS determining the step size, is preferable over the formulation of equation (6.9) in which ΔV determines the step size.

Reference 14 mentions the possibility of a gradient projection search in which the flutter speed is held constant and the total weight is reduced. The present authors consider this a more significant procedure from a practical point of view. As indicated in Reference 14 the formulas for this approach can be obtained from the constant total weight approach by interchanging the symbols V and M and changing the algebraic sign on ΔS .

Since the total mass M has a minimum value, only the equivalent of equation (6.8) will be given:

$$\left\{ \Delta P_i \right\} = \frac{-\Delta S}{\sqrt{\sum_{i=1}^n \left\{ \left(\frac{\partial M}{\partial P_i} \right)^2 + \lambda_1 \frac{\partial M}{\partial P_i} \frac{\partial V}{\partial P_i} \right\}}} \left\{ \frac{\partial M}{\partial P_i} + \lambda_1 \frac{\partial V}{\partial P_i} \right\} \quad (6.12)$$

where:

$$\lambda_1 = - \frac{\left[\frac{\partial M}{\partial P_i} \right] \left\{ \frac{\partial V}{\partial P_i} \right\}}{\left[\frac{\partial V}{\partial P_i} \right] \left\{ \frac{\partial V}{\partial P_i} \right\}} \quad (6.13)$$

which follows from the condition

$$\Delta V = \left[\Delta P_i \right] \left\{ \frac{\partial V}{\partial P_i} \right\} = 0 \quad (6.14)$$

Assuming a linear relationship between the elementary masses, m_i , and the design variables, P_i :

$$m_i = C_i P_i \quad (6.15)$$

equations (6.12) and (6.13) become:

$$\left\{ \Delta P_i \right\} = \frac{-\Delta S}{\sqrt{\sum_{i=1}^n \left(C_i^2 + \lambda_1 C_i \frac{\partial V}{\partial P_i} \right)}} \left\{ C_i + \lambda_1 \frac{\partial V}{\partial P_i} \right\} \quad (6.16)$$

$$\lambda_1 = - \frac{\left[C_i \right] \left\{ \frac{\partial V}{\partial P_i} \right\}}{\left[\frac{\partial V}{\partial P_i} \right] \left\{ \frac{\partial V}{\partial P_i} \right\}} \quad (6.17)$$

Letting, in addition, $C_i = 1$, i.e., choosing the elementary masses as design variables, these two equations become:

$$\left\{ \Delta m_i \right\} = \frac{-\Delta S}{\sqrt{n + \lambda_1 \sum_{i=1}^n \frac{\partial V}{\partial m_i}}} \left\{ 1 + \lambda_1 \frac{\partial V}{\partial m_i} \right\} \quad (6.18)$$

$$\lambda_1 = - \frac{\sum_{i=1}^n \frac{\partial V}{\partial m_i}}{\sum_{i=1}^n \left(\frac{\partial V}{\partial m_i} \right)^2} \quad (6.19)$$

The step size is determined by

$$|\Delta S|^2 = \sum_{i=1}^n \Delta m_i^2 \quad (6.20)$$

Equation (6.18) will now be discussed under the assumption that

$\sum \frac{\partial V}{\partial m_i} > 0$ and thus $\lambda_1 < 0$. This implies that $\left[\frac{\partial V}{\partial m_i} \right] \{1\} > 0$. The latter inequality indicates that a uniform increment of all the design variables increases the flutter speed. This is not an unreasonable assumption. Additionally it can be shown that $n + \lambda_1 \sum_{i=1}^n \frac{\partial V}{\partial m_i} \geq 0$ for any arbitrary set of values of $\frac{\partial V}{\partial m_i}$.

In view of these considerations equation (6.18) can be written as:

$$\left\{ \Delta m_i \right\} = -K_1 \left\{ \lambda_1 \frac{\partial V}{\partial m_i} + 1 \right\} \quad (6.21)$$

where $K_1 = \frac{\Delta S}{n + \lambda_1 \sum_{i=1}^n \frac{\partial V}{\partial m_i}} > 0$ and $\lambda_1 \frac{\partial V}{\partial m_i}$ can be larger or smaller

than 1.

The resizing column defined by equation (6.21) is a linear combination of the columns associated with the velocity and mass gradient search discussed in the previous sections. The algebraic sum of the two contributions tends to

increase the design variables with a larger value of $\frac{\partial V}{\partial m_i}$ and to decrease those with a smaller value of $\frac{\partial V}{\partial m_i}$, including those with a negative value of $\frac{\partial V}{\partial m_i}$. This agrees with physical reasoning. As defined in equation (6.18) the two contributions to $\left\{ \Delta m_i \right\}$ in equation (6.21) are in a fixed ratio. Due to nonlinearities this results in a drift in flutter speed due to a resizing step $\left\{ \Delta m_i \right\}$.

By writing

$$\left\{ \Delta m_i \right\} = \frac{\Delta V}{\sum_{i=1}^n \left(\frac{\partial V}{\partial m_i} \right)^2} \left\{ \frac{\partial V}{\partial m_i} \right\} - \bar{K} \left\{ 1 \right\} \quad (6.22)$$

an equivalent of equation (6.21) is formulated that allows the use of the method of Incremented Flutter Analysis (References 4 and 1) for determining a value of \bar{K} such that the velocity increment, ΔV , associated with the velocity gradient column is exactly cancelled.

In equation (6.21), K_1 determines the step size. In a follow-on paper (Reference 19) Rudisill and Bhatia describe a deterministic method of defining step size in the case that the gradient projection search is used to increase the flutter speed at constant total mass. The method is based on the assumption that the flutter speed is a nearly quadratic function of the design variables.

Since the total mass is a linear function of the design variables this method of determining the step size is not applicable to the gradient projection search at constant flutter speed. Choosing a step size then is a matter of experience and judgment.

6.2.4 Concluding Remarks - The preceding discussion deals with the basic resizing columns as defined by four different approaches. In any practical application, minimum size constraints must be taken into account. Conceptually this is a simple matter. Computer programming requires close attention to detail since a variety of potential minimum size constraints, some of them expected to occur infrequently, must be foreseen. Reference 14 does not go into detail on this.

The capability of the approaches presented in Reference 14 for structural optimization with a flutter constraint lies exclusively in the use of the gradient of the flutter speed with respect to the design variables. Repeated addition of structural material according to a distribution proportional to $\left\{ \frac{\partial V}{\partial m_i} \right\}$, followed each time by a rather indiscriminate removal of material to maintain the desired flutter speed, converges to a minimum mass design.

6.3 The Weight Gradient Method of Simodynes

In March of 1973, E. E. Simodynes presented a method for the optimization of structural weight at a specified flutter speed (Reference 15). The method employs the gradient of total weight with respect to $n-2$ design variables in a resizing algorithm to minimize structural weight while maintaining a constant flutter speed. Two of the design variables are dependent variables. The resizing column is such that a constant flutter frequency is also maintained, a characteristic which simplifies the formation of the required derivatives. The restriction on flutter frequency, however, represents an arbitrary constraint on the procedure which may result in weight penalties which can not be justified by the computational simplifications achieved. A modification of the method is proposed wherein the flutter frequency is permitted to vary, taking the place of one of the dependent design variables. In the following, the original method is discussed first, followed by a discussion of the modified procedure.

6.3.1 The Method of Simodynes - The optimization method presented in Reference 15 uses a resizing algorithm based on the gradient of total weight,

subject to flutter speed and flutter frequency constraints. As with most optimization methods, the stiffness and inertia terms are expressed in linear form:

$$[K] = [K_0] + \sum_{i=1}^n m_i [\Delta K_i] \quad \text{and} \quad [M] = [M_0] + \sum_{i=1}^n m_i [\Delta M_i] \quad (6.23)$$

The matrices $[K_0]$ and $[M_0]$ represent the stiffness and inertia of the fixed structure, and the matrices $[\Delta K_i]$ and $[\Delta M_i]$ represent stiffness and inertia increments per unit weight of the i^{th} design variable. The flutter equation for neutral stability is

$$\left([K_0] - \omega^2 [M_0] - \omega^2 [Q] + \sum_{i=1}^n m_i ([\Delta K_i] - \omega^2 [\Delta M_i]) \right) \{q\} = \{0\} \quad (6.24)$$

where $[Q]$ is a matrix of unsteady aerodynamic force coefficients.

Considering flutter frequency and flutter velocity to be specified, the fixed and variable coefficients of the flutter equation may be grouped as in equations (6.25) and (6.26), and the flutter equation expressed as in equation (6.27).

$$[A] = [K_0] - \omega^2 [M_0] - \omega^2 [Q] \quad (6.25)$$

$$[B_i] = [\Delta K_i] - \omega^2 [\Delta M_i] \quad (6.26)$$

$$\left([A] + \sum_{i=1}^n m_i [B_i] \right) \{q\} = 0 \quad (6.27)$$

Two of the design variables are now designated as the dependent design variables m_u and m_v , and derivatives of the flutter equation with respect to the independent design variables m_i formed as shown in equation (6.28).

$$[r][B_i]\{q\} + [r][B_u]\{q\} \frac{\partial m_u}{\partial m_i} + [r][B_v]\{q\} \frac{\partial m_v}{\partial m_i} = 0 \quad (6.28)$$

In this equation, $\{q\}$ is the flutter eigenvector of equation (6.27), and $[r]$ is the corresponding row eigenvector. Since the complex coefficients of equation (6.28) are known, the partial derivatives $\partial m_u / \partial m_i$ and $\partial m_v / \partial m_i$ are readily obtained. The weight of the structure is now expressed in equation (6.29), where W_0 is the weight of the fixed structure; the derivative of the weight with respect to each independent design variable is computed (equation (6.30)), and the gradient of total weight formed (equation (6.31)).

$$W = W_0 + m_u + m_v + \sum_{i=1}^{n-2} m_i \quad (6.29)$$

$$\frac{\partial W}{\partial m_i} = \frac{\partial m_u}{\partial m_i} + \frac{\partial m_v}{\partial m_i} + 1 \quad (6.30)$$

$$\{\nabla W\} = \begin{Bmatrix} \partial W / \partial m_1 \\ \partial W / \partial m_2 \\ \vdots \\ \partial W / \partial m_{n-2} \end{Bmatrix} \quad (6.31)$$

A resizing column of increments is then formed for a specific weight reduction, $-\Delta W$ (equation (6.32)), and new values of m_u and m_v are calculated using this column of increments and the partial derivatives $\partial m_u / \partial m_i$ and $\partial m_v / \partial m_i$.

$$\{\Delta m_i\} = \frac{-\Delta W}{[\nabla W]^T \{\nabla W\}} \{\nabla W\} \quad (6.32)$$

6.3.2 Discussion of the Method - As indicated in Reference 15, the principal feature of the method of Simodynnes is the weight gradient, and the resizing column derived from it. The computation of this weight gradient is particularly straightforward due to the constraints on flutter speed and flutter frequency imposed on the resizing procedure. In particular, the unsteady

aerodynamic parameters are constant throughout the optimization cycle, so that the flutter derivatives do not involve derivatives of the aerodynamic parameters.

This simplification, however, is not obtained without an associated disadvantage. The imposition of the frequency constraint on the optimization process results in an artificial constraint on the distribution of the design variable masses during the resizing procedure. An example of this may be seen in the numerical evaluations reported in Appendix A. The effect of this constraint is difficult to assess, since it depends to a great degree on the particular situation. For a configuration far from the final optimum distribution, with a frequency of the critical mode significantly different from the final value, the effect of the frequency constraint would be expected to be significant. For a configuration closer to the optimum distribution, with a critical mode frequency approximately equal to that of the final configuration, the effect would be correspondingly less. Reference 15 states that a fixed set of vibration modes is used throughout a complete optimization cycle, and it is concluded that the flutter frequency constraint is maintained throughout the optimization cycle also. In that case, at least one additional optimization cycle must be performed in order to assure that no effects of the constraint remain. The choice of dependent design variables also influences the magnitude of the constraint effect, since design variables having little influence on frequency would produce relatively greater distortions of the unconstrained distributions.

As with other methods employing an arbitrary step-size (e.g. Reference 14), this parameter must be established on the basis of judgment, intuition, experience or (probably) a combination of these. For the present method, step-size is determined by the weight reduction, $-\Delta W$, specified by the user. The choice of this parameter involves a compromise; too large a value of ΔW could result in unacceptably large excursions in flutter speed and frequency; too small a value of ΔW would result in an excessive number of resizing steps required to reach optimum.

Minimum size constraints are handled in a straightforward manner; when a design variable reaches minimum size, it is temporarily eliminated as a design variable. In subsequent steps, the derivatives are calculated for this design variable and it is reinstated as an active design variable if these calculations so indicate. Although not specifically stated in the reference, this is presumably true of the dependent design variables as well as the independent design variables. Here again, a poor choice of dependent design variables, requiring frequent shifting to other design variables during the optimization cycle, would result in an inefficient resizing process.

6.3.3 A Modification of the Method of Simodynes - As discussed earlier, the constraint on the flutter frequency provides a simplification in the computation of the required derivatives, but results in a nonoptimum weight increment implying a weight penalty that is difficult to predict. A modification is suggested which eliminates this frequency constraint. This is done by allowing the flutter frequency to become a dependent variable, taking the

place of one of the dependent design variables. Following a procedure similar to that of the original method, derivatives of the flutter equation are obtained after first expressing the flutter equation as a function of the reduced frequency, k . The result is equation (6.33), which is comparable to equation (6.28) of the original procedure. Equation (6.33) is solved for the

$$\begin{aligned}
 & - \left[r \right] \left[\Delta M_i \right] \left\{ q \right\} k^2 - \left[r \right] \left[\Delta M_u \right] \left\{ q \right\} k^2 \frac{\partial m_u}{\partial m_i} + \\
 & - 2 \left[r \right] \left[M \right] \left\{ q \right\} \frac{\partial k}{\partial m_i} + \left[r \right] \left[\Delta K_i \right] \left\{ q \right\} + \\
 & + \left[r \right] \left[\Delta K_u \right] \left\{ q \right\} \frac{\partial m_u}{\partial k} - \left[r \right] \left[\frac{\partial A(k)}{\partial k} \right] \left\{ q \right\} \frac{\partial k}{\partial m_i} = 0
 \end{aligned} \tag{6.33}$$

unknown parameters $\partial m_u / \partial m_i$ and $\partial k / \partial m_i$; the total weight is expressed in equation (6.34), and the derivative of the weight with respect to each

$$W = W_0 + m_u + \sum_{i=1}^{n-1} m_i \tag{6.34}$$

$$\frac{\partial W}{\partial m_i} = 1 + \frac{\partial m_u}{\partial m_i} \tag{6.35}$$

of the independent design variables is shown in equation (6.35).

The gradient of the weight and the column of increments are formed as before (equations (6.31) and (6.32)). Except for this modification in the weight gradient formation, the optimization cycle proceeds as in the original method.

6.3.4 Discussion of the Modified Method - The modification of Simodynes' method suggested here unquestionably results in a less approximate procedure. The frequency constraint, with the associated distortion of the resulting mass distribution, is removed. Not only does this eliminate an undesirable feature of the original method, but fewer optimization cycles should be required in order to reach a satisfactory approximation of the optimum distribution. As a result of the modification, however, some additional complication of the computational procedure is required. In solving the flutter equation to obtain the eigenvectors, a fixed matrix of unsteady aerodynamic coefficients can no longer be used throughout the optimization cycle. In addition, the derivatives of the aerodynamic parameters with respect to the reduced frequency, k , must be obtained. How troublesome these complications

are depends on the versatility of the computational system used. For the numerical evaluation of the modified method presented in Appendix A, the Lockheed p-k method of solving the flutter equation was used; this procedure has a built-in subroutine for interpolating matrices of aerodynamic coefficients. A similar routine was used to obtain the approximate derivatives of the aerodynamic parameters using a finite-difference procedure. Since these computational tools were readily available, the modifications resulted in no particular computational difficulty.

It should be noted that a certain similarity exists between the column of increments resulting from the present procedure and the column of increments resulting from the gradient projection search with constant flutter speed of Rudisill-Bhatia (Section 6.2.3). The expression for the column of increments of the present procedure is shown in equation (6.36) and from equation (A.6) of Appendix A it is seen that equation (6.37) is an equivalent expression.

$$\begin{Bmatrix} \Delta m_i \\ -\Delta m_u \end{Bmatrix} = \frac{-\Delta W}{[\nabla W] \begin{Bmatrix} \nabla W \end{Bmatrix}} \begin{Bmatrix} 1 + \frac{\partial m_u}{\partial m_i} \\ -\frac{\frac{\partial m_u}{\partial m_i} + \sum \left(\frac{\partial m_u}{\partial m_i} \right)^2}{\sum \frac{\partial m_u}{\partial m_i}} \end{Bmatrix} \quad (6.36)$$

$$\begin{Bmatrix} \Delta m_i \\ -\Delta m_u \end{Bmatrix} = \frac{-\Delta W}{[\nabla W] \begin{Bmatrix} \nabla W \end{Bmatrix}} \begin{Bmatrix} 1 - \frac{\partial V / \partial m_i}{\partial V / \partial m_u} \\ -\frac{\frac{\partial V / \partial m_i}{\partial V / \partial m_u} + \sum \left(\frac{\partial V / \partial m_i}{\partial V / \partial m_u} \right)^2}{-\sum \frac{\partial V / \partial m_i}{\partial V / \partial m_u} + \sum \left(\frac{\partial V / \partial m_i}{\partial V / \partial m_u} \right)^2} \end{Bmatrix} \quad (6.37)$$

Recognizing that the scalar premultiplier of the column is arbitrarily chosen, the increments for the independent design variables m_i can be made identical to those of equation (6.21) of Section 6.2.3 if the normalization factor $1/\partial V/\partial m_u$ is equal to $-\lambda_1$ of equation (6.21). It will be recalled that λ_1 is chosen so as to result in a flutter velocity increment, on a linear basis, equal to zero. In general, the normalization factor for the present procedure will not be equal to $-\lambda_1$, since it results from the choice of the dependent design variable. The (linear) flutter speed increment is

then held to zero by the increment in the dependent design variable defined in equation (6.37).

6.3.5 Assessment of the Method - The original Simodynes method is a simple, straightforward procedure which involves a minimum of computational difficulty in the generation of the flutter derivatives. The flutter frequency constraint might be a useful device in adopting existing routines for the solution of the flutter equation and generation of unsteady aerodynamic parameters for use in a flutter optimization procedure. In the development of an integrated design procedure, however, there would appear to be no clear advantage in retaining the frequency constraint. Instead, a modification of the procedure such as indicated here would be highly desirable.

6.4 An Interior Penalty Function Method

In seeking to evaluate a penalty function method, the initial task is one of defining both the method itself and the scope of the evaluation. As used here, the term penalty function method refers to any structural optimization technique in which penalty terms, which are related to the constraint equations, are added to an objective function to form a modified objective function, which is then minimized. For the purposes of this evaluation, however, only a single, representative procedure will be considered. The particular procedure chosen is an interior penalty function method and follows closely the method described in Reference 16, and is essentially identical to that used in the numerical evaluations presented in Appendix A. A brief description of this method is presented in the next section, preparatory to the discussion and evaluation which follow.

6.4.1 Description of Method - As indicated previously, the method considered here is based on the method described in Reference 16. In that procedure, a modified objective function, $P(m_i, r)$, is formed as shown in equation (6.38).

$$P(m_i, r) = W(m_i) + r \cdot \sum_{\ell} \frac{1}{g_{\ell}(m_i)} \quad (6.38)$$

The term $W(m_i)$ is the quantity to be minimized, or objective function, and represents the weight associated with the design variables m_i . The second term of equation (6.38) expresses the penalty terms as functions of the design constraints; r is a penalty function weighting factor. By means of this formulation, the problem of minimizing $W(m_i)$, subject to the constraints $g_{\ell}(m_i)$, is transformed to one of an unconstrained minimization of $P(m_i, r)$

using a SUMT (Sequence of Unconstrained Minimization Technique) approach (Reference 20 and 21). One such minimization is carried out for each successive reduction of the value of the penalty function weighting factor, r ,

until the minimized value of $P(m_i, r)$ is approximately equal to the corresponding value of $W(m_i)$.

The unconstrained minimization of $P(m_i, r)$ is accomplished by first generating a move-vector direction, then determining the minimum of $P(m_i, r)$ in this direction by means of a one-dimensional search. For determining this move direction, several direction-generating algorithms are available, the best known of which is the DFP algorithm (Reference 22) as used in the preliminary design procedure reported in Reference 9. Reference 16, however, uses a variation of Newton's method wherein the second derivatives of $P(m_i, r)$ are approximated, and this procedure will be used here. The second derivatives shown in equation (6.39) may be approximated by neglecting

$$\frac{\partial^2 P}{\partial m_i \partial m_j} = \frac{\partial^2 W}{\partial m_i \partial m_j} + r \cdot \sum_{\ell} \left(2 \frac{\partial g_{\ell}}{\partial m_i} \frac{\partial g_{\ell}}{\partial m_j} - g_{\ell} \frac{\partial^2 g_{\ell}}{\partial m_i \partial m_j} \right) g_{\ell}^{-3} \quad (6.39)$$

the second term of the summation, under the assumptions stated in Reference 16. In addition, the objective function, $W(m_i)$, is assumed to be a linear

function of the design variables, so that the first term of equation (6.39) is equal to zero. The second derivatives are then approximated as in equation (6.40), and an estimate for the column of design variable

$$\frac{\partial^2 P}{\partial m_i \partial m_j} \approx G_{ij} = 2r \cdot \sum_{\ell} \left(\frac{\partial g_{\ell}}{\partial m_i} \frac{\partial g_{\ell}}{\partial m_j} \right) g_{\ell}^{-3} \quad (6.40)$$

increments which will minimize $P(m_i, r)$ is formed as shown in equation (6.41).

$$\left\{ \Delta m_i \right\} = - \left[G \right]^{-1} \left\{ \frac{\partial P}{\partial m_i} \right\} \quad (6.41)$$

It should be noted that equation (6.41) would exactly define the required column of design variable increments if $P(m_i, r)$ were a quadratic function of the design variables and the exact second derivatives used. In the present case, however, equation (6.41) is used only to define the move direction for the one-dimensional search.

6.4.2 Discussion of the Method - In evaluating the penalty function method described here, four principal elements or characteristics can be discerned: the treatment of the constraints, the penalty term weighting factors, the direction generating algorithm and the one-dimensional search. In the following sections, each of these characteristics will be discussed, followed by an assessment of the method as a complete procedure.

The treatment of constraints is the principal distinguishing feature of the penalty function method, and this treatment provides a number of advantages. As used in the present procedure, the inequality constraints serve two important functions: 1) conditioning of a move vector such that it tends to avoid violation of the constraints, and 2) limiting of the move vector amplitude such that the constraints are not violated. The first of these functions is implemented through the direction generating procedure, and the second of these is a part of the one-dimensional search.

The treatment of the constraints as inequality constraints, resulting in the characteristics described above, can be a very powerful approach in structural optimization for flutter. One of the more obvious advantages is the fact that the inclusion of multiple flutter speed constraints causes little conceptual difficulty. Whether this advantage is of practical value is difficult to assess without further work. To include all flutter speed constraints (for several Mach numbers and airplane loading conditions) in the penalty term would be a large computational burden. Thus it would seem that only active constraints should be included in the penalty term and that separate program logic should be used to determine which constraints are active. Experience at the Lockheed-California Company, partly obtained during this study, indicates that multiple active flutter speed constraints may occur rarely. Probably the most important advantage resulting from this method of handling constraints is the fact that a large number and variety of constraints can readily be included in an automated procedure. A disadvantage, however, is the fact that derivatives of the constraint quantities with respect to the design variables must be obtained.

The handling of the penalty term weighting factors indicated in equation (6.38) exerts a significant influence on the resultant performance of the method. The treatment of these factors is a matter of judgment and depends on experience. Experience at NASA, Langley Research Center suggests an initial value for these weighting factors which makes the penalty terms approximately equal to the value of the objective function (Reference 23). The final value for these factors can be determined by establishing acceptable

values of the residual constraint inequalities and then specifying an allowable percentage of the modified objective function contributed by the penalty terms. Having thus established the range of the weighting factors, the appropriate reduction factor to be applied during each step is determined from the selection of the desired number of steps. Monitoring of the progress of the weight minimization, however, may lead the analyst to interrupt the optimization procedure and modify the penalty weighting factors. Specifically, optimization steps may continue after the number of steps on which the reduction factor is based are completed if the total weight progression indicates that a minimum weight has not been attained.

At each step in the optimization process, the weighting factors determine the extent to which the constraints influence the resultant move. If the weighting factors are large relative to the distance from the constraint, the design tends to move away from the constraint, into the feasible design space. If the weighting factors are small, the constraints exert little influence on the move and the design may move in the direction of the constraints. The behavior is dependent on the reduction factor applied to the weighting factors at each successive step, which is in turn related to the range of the weighting factors and the number of steps selected. A large number of steps (or a small reduction factor) can result in erratic behavior of the process due to the strongly repelling influence of the constraints. Too few steps, with the corresponding large reduction factors, may result in moves which impact one or more constraints before significant weight reductions have been accomplished.

The use of the approximate second derivatives in an adaptation of Newton's method results in a more efficient unconstrained minimization procedure than does the use of the DFP algorithm (Reference 22). Since this latter procedure requires a number of one-dimensional searches approximately equal to the number of design variables, and the number of such searches required with Newton's method is independent of the number of design variables, the advantage of Newton's method increases as the number of design variables increases. As indicated in equation (6.41) however, the matrix G must be inverted; this matrix is an $n \times n$ matrix where n is the number of design variables. Reference 16 states that this matrix is singular or ill-conditioned when the number of active constraints is small. To preclude the obvious difficulties which would otherwise result, a matrix \bar{G} is used in place of the matrix G , the elements of which are shown in equation (6.42).

In this equation, δ_{ij} is the Kronecker delta and a value

$$\bar{G}_{ij} = G_{ij}(1 + \epsilon \delta_{ij}) \quad (6.42)$$

of $\epsilon=0.01$ is found to be satisfactory for the systems evaluated thus far.

For each step in the optimization process, a one-dimensional search is conducted to determine the minimum of the modified objective function. The direction of the move vector is determined as previously described (equation (6.41)), and only the magnitude of the move vector is varied during

the search. The derivatives of the modified objective function need only be evaluated once during each step, but the modified objective function is evaluated for a number of values of the move vector magnitude sufficient to define the minimum of the function. These evaluations of the modified objective function require flutter speed solutions, stress analyses and/or other procedures appropriate to the determination of the penalty terms. Although the process of evaluating the acceptability of a design relative to the constraints is common to all optimization methods, the penalty function method, employing a one-dimensional search, requires at least three such evaluations per step, whereas methods which do not employ the one-dimensional search need only one such evaluation. These other methods, however, normally require a greater number of steps to achieve an acceptable level of optimization, and therefore usually require a larger number of flutter speed derivative determinations. Considering that determining the flutter speed derivatives requires determination of the flutter root and two characteristic vectors, it is estimated that in terms of computational operations (and thus cost) each step in the penalty function procedure, requiring the calculation of three flutter roots, is approximately equivalent to three steps of a procedure such as the modified Simodynes method described in Section 6.3. This trade-off must be taken into account in any comparative evaluation of methods, and must be determined for a realistic design procedure. The principal advantage of the one-dimensional search is that it provides a specific criterion for the determination of step size, resulting (usually) in greater weight reductions per step than obtained with methods employing arbitrary step size. The determination of step size by the present procedure can readily be implemented as a fully automated computational subroutine.

6.4.3 Assessment of the Method - Based on the description of the optimization method presented in Section 6.4.1 and the discussion of the characteristics of the method in Section 6.4.2, it is concluded that the penalty function method (as that term is used here) is an efficient optimization procedure possessing characteristics not found in other methods. The unique means of handling design constraints is, of course, the most noteworthy of these. This feature, along with the use of the one-dimensional search to determine step size, results in an optimization procedure which is particularly well suited to use in a completely automated routine. Detailed specifications for such a procedure should be relatively easy to develop. The principal element required for the use of this method, over and above the requirements of other methods investigated, is the determination of the derivatives of the constraint quantities (other than flutter derivatives) with respect to the design variables. Since it should always be possible to evaluate the constraint functions for specified values of the design variables, these derivatives may be obtained by finite difference techniques if no better method is available. In terms of computational efficiency, it is difficult to compare the penalty function method with the other four methods considered here. From the foregoing it is clear that the penalty function method does, in general, require more computations per step than do the Simodynes or Rudisill-Bhatia procedures. Whether or not the more efficient optimization step of the penalty function method overcomes that disadvantage can only be assessed in terms of a realistic design case.

6.5 A Method of Feasible Directions

The method of feasible directions is an approach to structural optimization using direct minimization of a constrained function. This is in contrast to penalty function methods, treated in Section 6.4, which convert the constrained design problem into a sequence of unconstrained minimizations of a modified objective function.

The method discussed here is based primarily on the method of Gwin and McIntosh (Reference 17), which is in turn a generalization of a method developed by Zoutendijk (Reference 24). Additional background material is derived from Vanderplaats and Moses in Reference 25. It should be noted that the discussion presented here is limited to those characteristics inherent to the feasible directions method itself; other particulars of the method presented in Reference 17 are not treated. Thus, such elements as the method of generating aerodynamic parameters, solution of the flutter equation and computation of flutter speed derivatives are considered to be separate from the method of feasible directions. These and other details of a complete optimization procedure are considered elsewhere in this report.

6.5.1 Description of Method - The method of feasible directions generates a sequence of design changes, each of which is both feasible (does not violate active constraints) and usable (reduces total mass). Each resizing direction is followed until a new constraint is violated, an active constraint is re-encountered or the total mass is minimized. The process can be visualized with the help of Figure 6-1, which reproduces Figure 6 of Reference 17. In this figure, the minimum size constraints are represented by the horizontal and vertical boundaries, the minimum flutter speed constraint by the curved boundary and the constant weight contours by the diagonal straight lines. The starting point for the process is at point A, which is on the flutter speed constraint boundary; the design proceeds in a direction away from the flutter speed constraint and in a direction of decreasing weight until a constraint boundary is encountered. At that point, a new direction is generated and a new move executed. This process is continued until a point B is reached which approximates an optimum design. Figure 6-2, also from Reference 17, illustrates the range of acceptable design directions starting from a point B on a general nonlinear constraint boundary.

For a constraint equation of the form given in equation (6.43), the condition that the direction is feasible is given by equation (6.44), where $\{\nabla h\}$ is the gradient vector of the constraint with respect to the design variables, m_i , and $\{S\}$ is the direction vector of the design variables.

$$h \leq 0 \quad (6.43)$$

$$\{S\} \{\nabla h\} \leq 0 \quad (6.44)$$

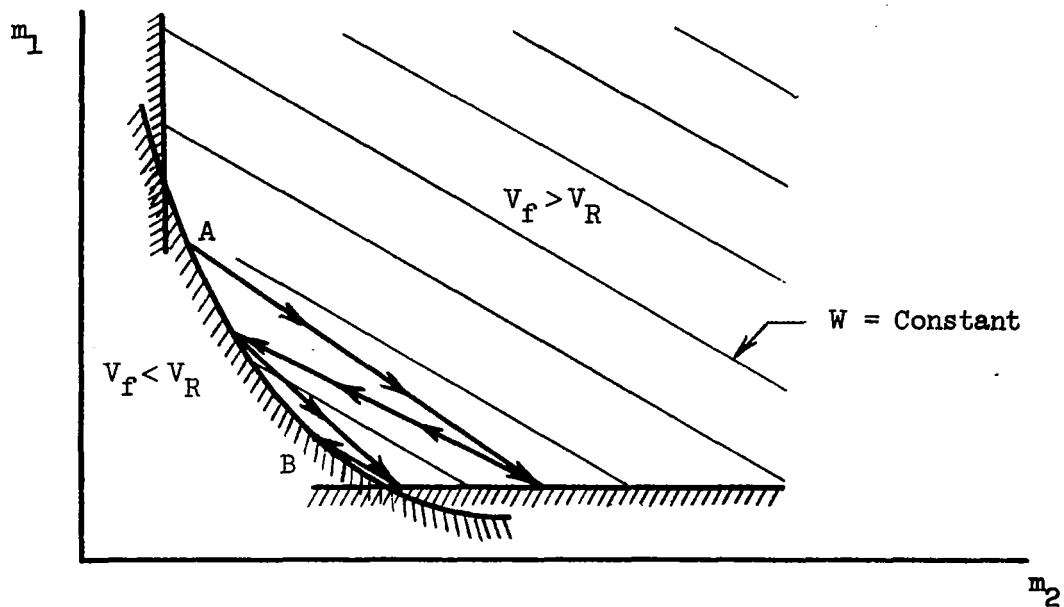


Figure 6-1: Hypothetical Design Space

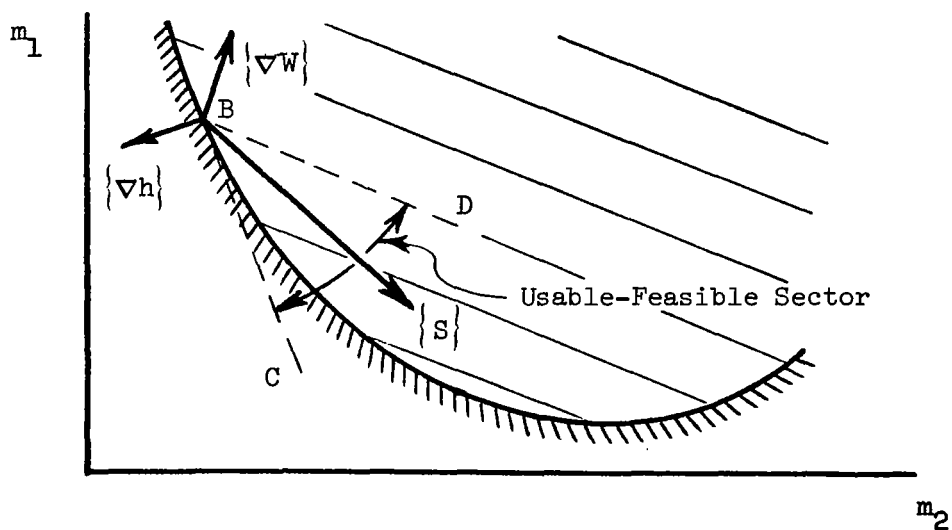


Figure 6-2: Direction-Finding Problem at a Constraint Boundary

In the present context, a feasible direction is one which does not violate an active constraint, as defined by a linear approximation of that constraint. The line segment BC in Figure 6-2 lies on the boundary of the feasible region. The condition that the direction is usable is given by equation (6.45), where $\{\nabla W\}$ is the gradient vector of the objective function, i.e., the total structural weight. Any direction between lines BD and BC in Figure 6-2 is then in the usable-feasible sector.

$$[S] \{\nabla W\} \leq 0 \quad (6.45)$$

The particular direction vector used in the present method is found such that a scalar β is maximized subject to the conditions expressed in equation (6.46). In this set of equations, θ is an adjustment factor which

$$(a) \quad [S] \{\nabla h\} + \theta \beta \leq 0$$

$$(b) \quad [S] \{\nabla W\} + \beta \leq 0 \quad (6.46)$$

$$(c) \quad \text{The length of } [S] \text{ is bounded.}$$

controls the direction of $[S]$ within the usable-feasible region; large values of θ force the direction away from the constraint and toward the usable boundary BD, while a value of $\theta = 0$ results in a move direction along line BC. For nonlinear constraints, such as a minimum flutter speed constraint, an intermediate value of θ is used which approximately bisects the usable feasible sector. Reference 17 suggests a value of $\theta = 1.0$. It will be shown, however, that the effect of any particular value of θ depends on the normalization of the constraint gradients and the weight gradient. For linear constraints, such as minimum size constraints, a value of $\theta = 0$ is used in order to produce a move direction parallel to the constraint boundary. The condition that the length of $[S]$ is bounded is usually accomplished by limiting the elements as shown in equation (6.47).

$$|s_i| \leq 1 \quad i = 1, 2, \dots, n \quad (6.47)$$

Once the value of θ is selected, the suboptimization problem indicated by equations (6.46) is transformed to a standard form and solved by means of the Simplex algorithm. Some of the details of this procedure are given in Reference 17, and a more comprehensive description may be found in Reference 26. The details of the Simplex algorithm will not be repeated here, but some aspects of the procedure are discussed in the following section. It should be noted that the Simplex algorithm was used to generate the direction vectors for the numerical evaluations presented in Appendix A of this report.

In applying the constraint conditions indicated by equation (6.46(a)), only those constraints which are considered to be active at a particular design point are included in the direction-finding process. A constraint is considered to be active if the design in question falls within a specified tolerance band ϵ of the constraint boundary. This constraint tolerance band is arbitrarily specified and may vary with type of constraint. As stated earlier, any particular design step continues in the specified direction until a constraint is violated; at that point, a correction step is taken back into the constraint tolerance band and a new direction generated.

6.5.2 Principal Characteristics of the Method - The principal distinguishing feature of the method of feasible directions is, as the name implies, the direction generating procedure. Some aspects of this procedure, and of the method of establishing the step size, are discussed in the following.

Some uncertainty exists as to the handling of the constraint tolerance band, ϵ . Reference 17 indicates that an appropriate constraint tolerance is established for each constraint or constraint type, and that when a constraint is violated, the step size is corrected such that the end-point of the step is midway in the tolerance band. Reference 25, however, recommends the use of a large tolerance band for the initial phases of the optimization so that the constraints will become active early in the optimization procedure and will remain active. Combining these two approaches would not appear to result in an efficient resizing procedure, since a large constraint tolerance band would prevent a close approach to any constraint. For the idealized test case of Appendix A, no particular advantage can be discerned in maintaining a large "pad" on the constraints; this would almost certainly result in an increased number of resizing steps for the weight reductions shown in Section A.6. It may be that such a procedure would be useful in the application of the method to more complex design problems as a means of avoiding convergence to localized minima, but it is felt that such usefulness would be rare in practical situations. Based on the idealized test case, it would appear that the constraint tolerance for minimum size constraints should be equal to zero. A minimum size constraint should not be active until the minimum size is reached, since otherwise no further reduction in that design variable would take place while the constraint remained active. For minimum flutter speed constraints, it would seem that the constraint tolerance band for defining an active constraint should indeed be fairly large. The flutter speed constraints then become active early in the design process, and are therefore effective in the efficient resizing of the design variables. The constraint tolerance band for determining the end-point of a particular design step, however, should be essentially equal to zero.

Assuming a reasonable value of the adjustment factor, θ , the next design step will be directed well into the feasible region, so that no useful purpose is served by originating the step any appreciable distance from the flutter constraint boundary.

For other types of constraint, the treatment of the constraint tolerance band can be based on similar reasoning.

As indicated earlier, the recommended adjustment factor for a minimum size constraint or other linear constraint is $\theta = 0$. This choice of adjustment factor results in subsequent design steps proceeding parallel to the constraint boundary if the design steps would otherwise result in continued reductions of the constrained design variable. For nonlinear constraints such as flutter speed constraints, a nonzero value of θ must be used in order to force the design direction away from the constraint boundary. Otherwise, any finite move amplitude would violate the flutter speed constraint. This characteristic of the parameter θ results in the term "push-off" factor being applied to it in some references. The most efficient value of the "push-off" factor depends on the particular structural optimization under consideration, but a value of $\theta = 1.0$ is usually chosen. Reference 17 states that this value produces a vector direction which approximately bisects the usable-feasible sector. Examination of equations (6.46) demonstrates, however, that the value θ which accomplishes this is dependent on the normalization of $\{\nabla h\}$ and $\{\nabla W\}$. A larger value of the normalized constraint gradient corresponds to a smaller value of θ , while a larger value of the normalized weight gradient corresponds to a larger value of θ . In the numerical evaluations described in Appendix A, the weight gradient was a unit column as a consequence of the choice of design variables, and the flutter speed constraint gradient was normalized such that the value of the average element is unity. A value of $\theta = 1.0$, in conjunction with that normalization, produced approximately the desired result. The more usual procedure of normalizing the gradients on the largest element would have produced approximately twice as much "push-off".

It is noted that Vanderplaats and Moses, Reference 25, recommend a variable "push-off" factor which is a function of the distance from the constraint boundary. In the present notation, this is expressed in equation

(6.48), where ϵ is the constraint tolerance, h_j^k is the value of the j^{th} constraint function at the k^{th} design step, and θ_0 is chosen as unity.

$$\theta_j = \theta_0 \left(\frac{h_j^k}{\epsilon} + 1 \right)^2 \quad (6.48)$$

Since the constraint functions have negative values anywhere within the feasible region, the effect of this treatment of the "push-off" factors would be to drive the design to the constraint tolerance boundary, where the value

of θ_j is equal to zero. Although this approach has the advantage of providing a uniform treatment of the "push-off" factors, it would appear that the resultant excursions of the design in and out of the constraint tolerance band might well offset any advantage derived from this approach.

The direction vector for feasible directions procedures is usually obtained by the Simplex method. As noted in Reference 25, "the resulting direction tends to point towards the corners of a hypercube in the design space ($S_i^q = 1$ or -1)."

The direction vectors obtained in the numerical evaluations of Appendix A certainly exhibit the indicated tendencies. Reference 25 goes on to propose an alternate direction vector formulation, based on imposing bounds on the total vector rather than the individual elements. Although this formulation has not been evaluated in detail, it is presumed that the resulting vector might be somewhat more efficient since it is subject to fewer constraints.

Returning to the original formulation, some elementary considerations of equations (6.46) and equation (6.47) will indicate the reasons for the resulting direction vector characteristics, and will suggest a method of generating the direction vector without the use of the Simplex algorithm. Referring to equations (6.46), assume for the moment that only the flutter constraint is active. It will be seen that maximum β can only occur when both equation (6.46(a)) and equation (6.46(b)) are equal to zero. If now the value of θ is taken as unity, equation (6.49) results when β is maximum.

$$[S] \left\{ \left\{ \nabla W \right\} - \left\{ \nabla h \right\} \right\} = 0 \quad (6.49)$$

Returning to equations (6.46) and observing that the elements of $\{\nabla h\}$ are generally negative, it is not difficult to see that the addition of weight to the design variable having the greatest increase in flutter speed per pound, and removing weight from the design variable having the least increase in flutter speed per pound, tends to increase β . The restriction on the

size of the elements S_i expressed in equation (6.47) limits these elements

to $+1$ and -1 , respectively, while the objective of maximizing β insures that these limits will be reached. Continuing the same line of reasoning, it will be seen that the resulting direction vector must have $+1$ elements for the design variables with the higher flutter speed derivatives, -1 elements for the design variables with the lower flutter speed derivatives, and one element of intermediate value in order to satisfy equation (6.49). If one or more design variables in the negative group is limited by an active minimum sizing constraint, the corresponding elements in the direction vector are set to zero and equation (6.49) balanced as before. It should be noted that the ranking of the flutter derivatives must be on the basis of rate of change of flutter speed per pound of design variable, which is the ratio of the elements of $\{\nabla h\}$ to the elements of $\{\nabla W\}$. During the course of the numerical evaluations presented in Appendix A, it was found that a move vector direction

identical to that generated by the Simplex method could be derived by the procedures described. When there are two active flutter constraints, V_1 and V_2 , the values of $\frac{\partial V_1}{\partial m_1} + \frac{\partial V_2}{\partial m_2}$ determine which elements of $[S]$ are +1 and -1; two equations similar to (6.49) will determine two elements of $[S]$ that have intermediate values. This was also demonstrated, numerically, to lead to the same $[S]$ as the one generated by the Simplex method. It is expected this approach can be expanded to more active flutter constraints. It is suggested that it is more straightforward and may be a more economical means of generating the direction vector.

Several methods of step-size selection are proposed in Reference 17, the simplest of which is to continue to increase step size in the prescribed direction until a constraint violation occurs. At this point, a corrected distance to a point within the constraint tolerance band is found by linear interpolation of the appropriate constraint function values.

As mentioned earlier, it should be relatively simple to determine the step size which would terminate exactly on the boundary of the nearest linear constraint, and in view of the fact that no further reduction of that design variable will take place while the constraint is active, no correction into the tolerance band would appear to be required. A more direct procedure might then be to determine the step-size to the nearest linear constraint and then check for violation of other constraints using that step-size. If constraint violations resulted, the step-size would be reduced until the critical constraint was just satisfied. For minimum flutter speed constraints, the use of some form of Incremented Flutter Analysis to solve for the step-size necessary to satisfy the flutter constraint should result in a substantial improvement in the step-size search procedure.

6.5.3 Assessment of the Method - The method of feasible directions is similar to the penalty function methods in that the constraint functions influence both the direction of the design step and the step-size. Full automation of the method is documented in Reference 27. A wide range of constraint types can be accommodated, the only requirement being that it must be possible to evaluate both the constraint functions and constraint gradients for each design step. In concept, multiple flutter constraints can be included, although there may be practical difficulties to be overcome outside the optimization proper.

The move direction vector resulting from the usual form of the equations appears to be rather crude, and it is felt that an alternate procedure, such as suggested in Reference 25, might result in a more efficient move vector. The treatment of constraint tolerances and determination of step-size seem overly complicated, and significant improvements in these areas should be possible.

Overall, the method seems to be quite representative of the direct methods of flutter optimization and competitive with other methods evaluated. The numerical evaluations reported in Appendix A show the method to be much better behaved than would be indicated by consideration of the move vector, and the rate of convergence demonstrated on the simplified test case is quite satisfactory. As a result, the method of feasible directions must be considered a strong candidate for further evaluation in a realistic design environment.

6.6 An Optimization Method Using Incremented Flutter Analysis

Incremented Flutter Analysis was conceived as a method for efficiently determining design parameters for external stores satisfying a predetermined flutter speed requirement (Reference 4). Its capability of determining the value of general design variables such that the flutter speed has a given value makes it attractive as a tool in optimization with flutter constraints. A study was therefore initiated to assess the usefulness of Incremented Flutter Analysis as a tool in an optimization procedure. It was used in a heavily interactive approach to optimization for flutter, using the Computer Graphics system. The resulting method of optimization was demonstrated on a simulated design problem based on a subsonic transport wing (Appendix A, Section A.7) and was also used on an actual design problem (arrow wing supersonic transport).

During this development, the method of Incremented Flutter Analysis was generalized to be consistent with the needs in a complex optimization program (Reference 1).

A breadboard prototype of an automated computer program was developed and demonstrated on the same simulated design problem.

In the following, the main features of the program and its present form are presented based on data in Reference 18.

6.6.1 Main Features - The optimization method is a resizing routine that minimizes total mass while maintaining the flutter speed exactly at a required value.

Key features of the method are that the resizing column is allowed to change direction, without the need to recalculate flutter speed derivatives, during a one-dimensional minimization process in which the value of a scalar α is determined that minimizes the total mass.

In the methods of optimization discussed in the preceding sections, a column of design variable increments (resizing column) $\{\Delta m_i^k\}$ is defined as:

$$\{\Delta m_i^k\} = \alpha^k \{d^k\} \quad (6.50)$$

where $\{d^k\}$ defines a direction in design variable space and the scalar α^k a magnitude. The resized design is related to the original design by

$$\{m_i^{k+1}\} = \{m_i^k\} + \{\Delta m_i^k\} \quad (6.51)$$

In this method the resizing column is

$$\{\Delta m_i^k\} = \{d^k(\alpha^k)\} \quad (6.52)$$

i.e., the direction of $\{\Delta m_i^k\}$ is a function of the scalar α . The scalar α is discussed in Section 6.6.2. During one resizing cycle, i.e., for one value of the superscript k , one set of derivatives of the flutter speed with respect to the design variables, $\frac{\partial V}{\partial m_i}$, is used. The column matrix $\{d^k(\alpha^k)\}$ is a function of $\frac{\partial V}{\partial m_i}$ as well as sizing constraints. During the one-dimensional minimization of the total mass with α as a variable, the method of Incremented Flutter Analysis is used to maintain the flutter speed exactly at the desired value.

6.6.2 Present Form of Program - The resizing column $\{\Delta m_i\}$ is defined as the sum of a basic resizing column $\{C_i\}$ and an adjustment column $\delta\{\Delta_i\}$.

The elements of $\{C_i\}$ satisfy the equation

$$\left[\frac{\partial V}{\partial m_i} \right] \{C_i\} = 0 \quad (6.53)$$

and thus would correspond to a zero change in flutter speed if V were a linear function of the m_i 's. The scalar δ of the adjustment column $\delta\{\Delta_i\}$ is determined such that

$$\{\Delta m_i\} = \{C_i\} + \delta\{\Delta_i\} \quad (6.54)$$

results exactly in a zero change in flutter speed.

In the following the notation $\frac{\partial V}{\partial m_i} = V_{mi}$ is used.

The design variables are divided in an R group and a Q group, such that

$$\left(V_{mi} \right)_{i \in Q} > \left(V_{mi} \right)_{i \in R} \quad (6.55)$$

The division between the R and Q group lies within the positive range of V_{mi} . Thus all design variables for which $V_{mi} < 0$ are in the R group. The largest V_{mi} is identified by $i = m$ and the smallest positive V_{mi} by $i = sp$. The division between the R and Q groups is defined by the inequality

$$\frac{V_{mm}/V_{mi} - 1}{V_{mm}/V_{msp} - 1} \geq VR \quad (6.56)$$

where VR is an empirical value. It was found, by numerical experimentation, that $VR = 0.3$ is an acceptable value for the test case reported in Appendix A. For $i = m$ the left hand side equals zero; for $i = sp$ it equals 1. Thus the design variables for which the inequality (6.56) is satisfied belong to the R group.

The algorithm in Reference 18 is based on removing mass from each design variable in the R group that is not at minimum size. Each mass removal is

individually coupled with a change of all design variables in the Q group such that equation (6.53) is satisfied. Mass removal from a design variable in the R group for which $V_{mi \in R} > 0$ requires addition of mass in the Q group. If $V_{mi \in R} < 0$ mass removal in the R group is compensated by mass removal in the Q group in order to satisfy equation (6.53).

The column matrix $\{C_q^{(r)}\}$ is the change in the design variables in the Q group due to removal of mass from design variable r in the R group. In Reference 18 the distribution

$$\frac{C_q^r}{C_m} = \frac{V_{mq}}{V_{mm}} \quad (C_m > 0) \quad (6.57)$$

is used if mass is to be added to the Q group ($C_m > 0$ and $V_{mi \in R} > 0$) and

$$\frac{C_q^r}{C_m} = \frac{V_{mm}}{V_{mq}} \quad (C_m < 0) \quad (6.58)$$

if mass is to be subtracted from the Q group ($C_m < 0$ and $V_{mi \in R} < 0$).

Equation (6.57) expresses that more is added to the design variables with the higher values of V_{mi} . Equation (6.58) expresses that more is subtracted from the design variables with the lower values of V_{mi} .

The distribution for removal of mass in the R group used in Reference 18 is:

$$C_r = - \left(1 - \frac{V_{mm}}{V_{mr}} \right)^2 C^* \quad (6.59)$$

The foregoing leads to a basic resizing column that is the sum of two columns that do not "overlap" and thus can be written as one column

$$\{C_i\} = \{C_q\} + \{C_r\} = \left\{ \frac{C_q}{C_r} \right\} \quad (6.60)$$

The column matrix $\{C_r\}$ is given by equation (6.59); $\{C_q\}$ by:

$$C_q = -C_r \left(\sum_r \frac{V_{mr} V_{mq}}{\sum_q V_{mq}^2} \right)_{r \in V_{mr} > 0} - C_r \left(\sum_r \frac{V_{mr}}{n_q V_{mq}} \right)_{r \in V_{mr} < 0} \quad (6.61)$$

where n_q is the number of design variables in the Q group.

Equations (6.59) and (6.61) define the basic resizing column $\{C_i\}$ with the scalar C^* defining a magnitude.

If the value of C^* is such that equation (6.59) leads to the violation of minimum size constraints, that equation is only used for the elements that do not violate the size constraints. The other elements are given values corresponding to the minimum size constraints.

Violation of a minimum size constraint by design variables in the Q group is expected to be infrequent. The program, however, has provisions to guard against such violation.

The distribution of the adjustment column $\delta \{\Delta_i\}$ is discussed in Reference 18. In the numerical example in Appendix A, $\Delta_i = 0$ except at $i = m$ (maximum $\frac{\partial V}{\partial m_i}$). It is suggested, however, that $\Delta_i = C_i$ for $i \in Q$ and $\Delta_i = 0$ for $i \in R$ is expected to be a better choice.

The value of the scalar δ is determined by means of Incremented Flutter Analysis. The determinantal flutter equation (according to equation (3.4)) is written as:

$$D\left\{(\gamma + i)k, g, V, \rho, m_i^k + C_i, \delta \Delta_i\right\} = 0 \quad (6.62)$$

In equation (6.62), $\gamma = 0$, $g = 0$, V and ρ have given values; m_i^k are the values of the design variables at the beginning of the current resizing step; C_i satisfies equation (6.53). Equation (6.62) is solved by two-dimensional Regula Falsi for k and δ . The complete resizing column is then determined by equation (6.54).

If no size limitations become active, simple substitution of equation (6.59) into equation (6.61) shows that all elements of the resizing column $\{C_i\}$ are proportional to C^* . Different values of C^* can be assumed; $\{\Delta m_i^k\} = \{C_i\} + \delta\{\Delta_i\}$ can be computed and $M = \left[1\right] \left\{m_i^k + \Delta m_i^k\right\}$ can be computed as a function of C^* . This is, in principle, the one dimensional minimization that determines the direction as well as the magnitude of the resizing column. The total mass M does have a minimum due to nonlinear effects, which are taken into account by means of Incremented Flutter Analysis.

When size limitations are active not all elements of $\{C_i\}$ are proportional to C^* and nonalgebraic operations are needed to obtain M as a function of C^* . The logic for this is presented in Reference 18.

To facilitate initiation of the one dimensional minimization, a variable $\alpha = - \sum_R C_R$ is used. It is the total mass removed from design variables in the R group and has a very simple relation to C^* . The quantity α has a simple physical meaning, independent of the number of design variables. An initial value can be chosen as a fraction of the total mass represented by the design variables.

The numerical example of Reference 18 is presented in the Appendix as Table A-11. It suggests that the method is very powerful in reducing the total mass by a large fraction (>80%) of the difference between the current total mass and the minimum mass in a single step.

Although it is not essential to the overall method, it should be noted that the program as described in Reference 18 uses the method of Incremented Flutter Analysis to determine the values of $\partial V / \partial m_i$ by means of a finite difference approach (Reference 1).

6.6.3 Concluding Remarks - The approach taken in this method is distinctly different from each of the other methods discussed, although it contains elements of several of these methods. One distinguishing feature is that the flutter speed is held exactly at the constraint value. In fact, this feature is used to include the nonlinear character of the flutter speed as a function of the design variables and makes it possible to do a one-dimensional minimization of the objective function itself, rather than of a modified objective function as in the penalty function method.

Another distinguishing feature is the departure from the traditional distributions in the resizing column, which are largely based on the gradient of the velocity and the total mass. It has been demonstrated that empirically generated distributions can lead to rapidly converging optimization procedures.

A third feature is the use of Incremented Flutter Analysis. It is not considered advantageous to use Incremented Flutter Analysis to determine the derivatives of the flutter speed by the finite difference method. However, Incremented Flutter Analysis, as executed with the help of the two-dimensional Regula Falsi method for solving two nonlinear equations with two unknowns, is an efficient method for keeping the flutter speed exactly constant. This use of Incremented Flutter Analysis could be used advantageously in some of the other methods of optimization: keeping the flutter speed exactly constant facilitates the observance of a convergence criterion for minimum total mass since side effects due to drift of the flutter speed are avoided.

6.7 Comparison of Optimization Methods

6.7.1 General - In comparing the optimization methods discussed in Sections 6.2 - 6.6, many differences in procedural detail are apparent. Specifically, differences in generating the distribution and magnitude of the resizing column can be recognized. Of these two, however, the more basic difference relates to the determination of the magnitude of the resizing column, or step size. Several of the methods - the weight gradient optimization of Simodynes (Section 6.3) and the velocity gradient, mass gradient and gradient projection methods of Rudisill-Bhatia (Section 6.2) - employ arbitrary step sizes. In contrast, the penalty function method (Section 6.4), the method of feasible directions (Section 6.5) and the method incorporating Incremented Flutter Analysis (Section 6.6) all make use of a step size determined by well defined criteria. In the following, these two groups are discussed separately and then a candidate resizing procedure is synthesized, based on the analytical and numerical evaluations conducted thus far.

6.7.2 Arbitrary Step-Size Procedures - The resizing procedures employing an arbitrary step size are characterized by a well-defined resizing cycle which is usually simple and straightforward. In general, one flutter solution (with two characteristic vectors) and one set of flutter derivatives are required for each step. The procedures based on constant flutter speed (Simodyne's weight gradient and the gradient projection search of Rudisill-Bhatia) rely on a linearization, based on the flutter velocity derivatives, to hold the flutter speed constant. As a result, the actual flutter speed tends to drift (downward, in most practical resizing exercises) and must periodically be corrected. It is this tendency which effectively limits step-size, since large excursions from the required flutter speed are undesirable. Rather than attempting to maximize step-size, however, a moderate step-size is chosen, based on experience and engineering judgment, and the attendant penalty of an increased number of steps is accepted.

In terms of specific procedures, the Rudisill-Bhatia methods should be somewhat more efficient than the weight gradient method of Simodynes. As indicated in Section 6.3, the frequency constraint imposed by this latter procedure results in a degree of approximation which probably cannot be

justified on the basis of the resulting simplifications. In the modified procedure (Section 6.3.3), this frequency constraint is removed and the resulting procedure is shown to be similar to the gradient projection search of Rudisill-Bhatia. In comparing these two procedures, the Rudisill-Bhatia approach has a significant advantage in that it does not require the selection of a dependent design variable, and thus eliminates the resulting influence of this choice on the performance of the procedure. The use of the flutter velocity derivatives as developed by Rudisill-Bhatia, rather than the normalized derivatives of Simodynes, has some advantage in a procedure which incorporates a nonzero flutter velocity increment. This would be the case when using the flutter velocity gradient to define a design variable distribution to increase the flutter speed of an initially deficient system, or to make small flutter velocity corrections during the resizing cycles. For use in constant flutter velocity procedures, however, it is shown in Section A.3.3, Appendix A, that the two forms of the derivative may be used interchangeably.

The two most useful procedures employing arbitrary step size would then appear to be the velocity gradient search and the gradient projection search of Rudisill-Bhatia. As envisioned in Section 6.7.4, this former procedure would not be implemented using an arbitrary step-size. The most useful application of the velocity gradient search procedure is in increasing the flutter speed of a flutter-deficient system to the required flutter speed in a nearly optimum manner. It is shown in Section A.4.1, however, that the use of a single step to accomplish this is probably the most effective procedure. As a consequence, it seems reasonable to use the Incremented Flutter Analysis technique (cf. page 43) to determine the step-size necessary to satisfy the flutter speed constraint exactly. The gradient projection search could also be improved by a modification of the resizing column such that the mass gradient component is replaced by the reciprocal of the flutter speed derivatives, resulting in a resizing column defined by equations (A.3), (A.4) and (A.5) of Appendix A. A comparison of Tables A-5 and A-8 of Appendix A indicates that, at least for the idealized test case evaluated there, a significant increase in efficiency results from this modification.

6.7.3 Defined Step-Size Procedures - In resizing procedures employing a defined step-size, an attempt is made to maximize the step-size so as to derive the maximum benefit from a single resizing step. This maximum step-size is determined by a well-defined set of criteria, usually involving the condition of the current design with respect to the design constraints. Evaluating these criteria usually involves the determination of the flutter speed, along with other constraint conditions, at several points along the move path during one step. In contrast to the arbitrary step-size procedures, then, the defined step-size procedures normally result in a greater mass reduction for each set of flutter derivatives calculated, but at the expense of a greater number of required flutter solutions.

Each of the three defined step-size procedures discussed in the preceding sections has distinct characteristics, and meaningful comparisons are difficult to make. Some general observations are possible, however, allowing some tentative conclusions to be drawn.

The penalty function procedure (Section 6.4) is perhaps the most versatile of the three procedures considered. The treatment of constraints is straightforward, making automation of the procedure particularly simple. A step is terminated before a constraint violation occurs, but the constraints are continuously active and exert some influence on the direction of each resizing step. The efficiency of the method is to a significant degree dependent on the handling of the penalty term weighting factors associated with the constraints (equation (6.38)), so that some judgement and experience are both required in the selection of these factors. A more troublesome difficulty might be encountered in the use of the direction generating algorithm based on Newton's method. As shown in equation (6.41), the matrix of coefficients of the design variable second derivatives must be inverted, and it is indicated in Reference 16 that this matrix may be singular or ill-conditioned. Although means to avoid this problem are suggested, computational difficulties may still arise in practical design tasks involving large numbers of design variables.

The method of feasible directions (Section 6.5) provides a treatment of constraints which is different from that of either of the other two methods considered here. For the flutter speed constraint (and presumably other nonlinear constraints) the approach appears to be quite satisfactory. It is similar to that of the penalty function method, in that the "push-off" factor can be considered as analogous to the penalty weighting factors of that procedure. In the feasible directions method, however, the move direction is restricted to the usable as well as feasible region. As a consequence, each move must reduce the objective function (total mass) as well as avoid a violation of the design constraints. In contrast, the penalty function move must reduce the modified objective function, but not necessarily the objective function itself. For linear constraints, such as minimum sizing constraints, the feasible directions method is formulated such that the constraints are not active until a constraint violation occurs, and the resizing step is terminated at that point. Prior to the constraint violation, the constraint exerts no influence on the move direction and therefore such constraint violations are normal occurrences. For the idealized test case evaluated in Appendix A, this characteristic did not appreciably degrade the efficiency of the procedure; only four sizing constraints became active during the optimization, and a significant weight reduction resulted from each step. In a more realistic design case, with a large number of minimum size constraints, it is anticipated that a significant number of short, ineffective moves would result from sizing constraint encounters. Once a linear constraint becomes active, however, the constraint is incorporated in the move direction so that subsequent moves take place along the constraint boundary. The conditions imposed on the direction of the move vector result in design variable increments that, in general, are equal in magnitude, are positive for the design variables with the higher values of $\partial V/\partial m$ and are negative for the design variables with the lower values of $\partial V/\partial m$. Thus there appears a lack of differentiation between the design variable increments with equal algebraic sign. It should be noted, however, that the results of the numerical evaluations described in Appendix A do not substantiate this lack of efficiency.

The procedure incorporating the use of Incremented Flutter Analysis in a formalized resizing procedure (Section 6.6) utilizes a concept to define step-size which is significantly different from that of either of the other two methods discussed here. In principle, a distribution of design variable increments is defined which reduces total mass and which, on a linear basis, produces no change in flutter speed. Since flutter speed is not a linear function of the design variables, however, any finite value of this incremental distribution will produce (in all practical cases) a decrease in flutter speed. For several values of the incremental distribution, the value of an adjustment increment is determined by the Incremented Flutter Analysis technique which brings the flutter speed exactly back to the required value. For some value of the incremental distribution the total mass will be a minimum; this point defines the end of the step. By this means, the nonlinearities in the flutter constraint are explicitly accounted for, and the maximum mass reduction for a given set of values of the flutter speed derivatives is achieved. In the process of determining the step-size associated with the minimum mass, minimum size constraints are enforced as necessary. To that extent, the direction vector of the design variables is a function of the move amplitude, the direction conforming to the size constraints.

As presented in Reference 18 and as used in the numerical evaluations of Appendix A, the method described in Section 6.6 differs in two other respects from the other methods evaluated. The flutter derivatives are obtained through the use of Incremented Flutter Analysis in the form of increments in each individual design variable required for a reference change in flutter speed. The results of the numerical evaluation indicate that this finite difference form of the flutter derivatives results in values comparable to those obtained from the analytic form, and that the two forms may be used interchangeably. It is recognized, however, that the use of this procedure for obtaining the flutter derivatives - which requires the equivalent of a flutter solution for each design variable - is not efficient for a practical design task involving a large number of design variables. In such a case, the use of the analytic form of the derivative would be more economical. The other area in which this method differs from those previously discussed is in the formation of the resizing move vector. The separation of the design variables into two groups, those with higher values of $\partial V/\partial m$ and those with lower values of $\partial V/\partial m$, is empirical, and it is not clear that the criterion used in Section 6.6 would be efficient in all cases. The distribution of the increments for the design variables with the higher values of $\partial V/\partial m$ is proportional to the velocity gradient, but the distribution for the design variables with the lower values of $\partial V/\partial m$ is a second order function of the reciprocals of the velocity derivatives. The results of a numerical evaluation using the move vector of Section A.3.3, Appendix A, in place of the move vector described in Section 6.6 indicate that the efficiency of the two move vectors is approximately equal, at least in terms of the idealized test case of Appendix A. In view of this, it is considered that the more complex move vector presented in Section 6.6 is not justified on the basis of present results.

6.7.4 Formulation of a Resizing Procedure - Based on the evaluations presented in this section and the results of the numerical evaluations presented in Appendix A, a resizing procedure can be formulated which will result in an improved performance over that of any of the specific methods discussed. It is recognized that the evaluations of the resizing procedures are not complete; any promising procedure must be further evaluated in terms of a realistic design task in order to arrive at firm conclusions. In particular, further evidence must be obtained to determine the relative efficiency of the arbitrary step-size and defined step-size procedures. For the time being, however, it will be premised that the problems associated with a practical design task will dictate the use of a defined step-size procedure. These problems, some of which are discussed in Section 7, would seem to indicate that the greatest possible mass reduction should be obtained for each step, since the structural reanalysis required per resizing step may be much more extensive than is generally recognized.

As qualified by the preceding paragraph, the preferred resizing procedure may be described in terms of the following characteristics:

Flutter speed derivatives are of the analytic type, calculated by the method of Rudisill-Bhatia (Section 6.2), possibly generalized by taking into account the derivatives of the vibration modes with respect to the design variables.

The initial resizing step is one which increases the flutter speed of the flutter-deficient design to the required value. This is done in a nearly-optimum manner by the addition of design variable increments distributed according to the velocity gradient. The total required flutter speed increment is obtained in a single step, and a form of Incremented Flutter Analysis is used to determine the magnitude of the step required to satisfy the flutter constraint exactly.

Subsequent resizing steps are performed at constant flutter speed, using the technique of minimization of the objective function (mass) described in Section 6.6 in order to determine the step-size. As discussed in that section, the primary distribution of design variable increments is such as to produce zero flutter velocity change on a linearized basis. Incremented Flutter Analysis is then used to determine the magnitude of an adjustment column of increments required to maintain the actual flutter speed constant. The total mass of the design variables, including both the primary and adjustment distributions, is determined as a function of step-size, and the step-size corresponding to minimum mass chosen. Using this configuration as a starting point, the resizing cycle is repeated.

The sizing constraints are satisfied in the manner of Section 6.6, with the direction of the move vector being modified as constraints are encountered during the minimization of the objective function. Note that the flutter constraint is satisfied at each substep of the minimization.

The move vector for design variables corresponding to positive values of $\partial V / \partial m$ is based on the combination of the velocity gradient and the

negative reciprocals of the flutter derivatives shown in Section A.3.3. For design variables corresponding to negative values of $\partial V/\partial m$, modified distributions will be used, some options of which are discussed in Reference 1.

A summary discussion of this procedure and an example of numerical results are presented in Reference 28.

7. CONSIDERATIONS RELATED TO A REALISTIC DESIGN ENVIRONMENT

In the literature on optimization with flutter constraints, the methods presented are illustrated with examples of varying complexity. One example in Reference 16 is based on 156 structural degrees of freedom and 23 design variables. The example in Reference 29 is based on 150 degrees of freedom and 100 design variables. As much as these numbers surpass the corresponding numbers in earlier examples in the literature, they fall short of what may be encountered in a realistic design environment. Thus, problems that may result from such an environment remain unexposed. During the present work, several aspects of dealing with an actual design have been examined. They are discussed in the following sections, together with other aspects to which little or no attention could be given.

7.1 Structural Model

The mathematical model representing the structure obviously is an important element of structural optimization with flutter constraints. Finite element structural models with thousands of elements and a corresponding number of nodal displacements as degrees of freedom are used for stress and stiffness analysis of a given structure. A duplication of effort can be avoided if the same structural model can be used for flutter optimization.

It should be noted that for a flutter analysis, or a loads analysis including aeroelastic effects, the refinement of a multi-thousand element structural model is not required. For the current type of subsonic transports, a relatively simple beam model suffices for flutter. For supersonic transports, however, as are in existence and projected for the future, a simple beam model is inadequate and a finite element model must be used. This implies that methods of optimization for flutter must be able to handle finite element type structural representation.

The typical structural model for stress analysis has a number of degrees of freedom that exceeds what at present seems practicable for the repetitive vibration analysis expected in a flutter optimization program. A reduced number of degrees of freedom can be obtained by using a stiffness matrix of

reduced size in the vibration analysis or by generating a coarser finite element model for aeroelastic analyses, which may or may not require further size reduction.

The common approach to coordinate reduction is one in which coordinates to be eliminated are assumed to have zero loads. This is often called static reduction in contrast with the approach of Reference 30; which can be called dynamic reduction, since the reduction is a function of the frequency. If the basic stiffness matrix $\begin{bmatrix} K_b \end{bmatrix}$ is partitioned as indicated in equation (7.1), the reduced matrix, $\begin{bmatrix} K_r \end{bmatrix}$, is given by equation (7.2).

$$\begin{bmatrix} K_b \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \quad (7.1)$$

$$\begin{bmatrix} K_r \end{bmatrix} = \begin{bmatrix} K_{11} \end{bmatrix} - \begin{bmatrix} K_{12} \end{bmatrix} \begin{bmatrix} K_{22} \end{bmatrix}^{-1} \begin{bmatrix} K_{21} \end{bmatrix} \quad (7.2)$$

If the incremental stiffness due to increasing the design variable β_i a unit amount is $\begin{bmatrix} \Delta K_i \end{bmatrix}$, the basic stiffness matrix, as a function of the β_i 's is:

$$\begin{bmatrix} K_b(\beta_i) \end{bmatrix} = \begin{bmatrix} K_b(0) \end{bmatrix} + \sum \beta_i \begin{bmatrix} \Delta K_i \end{bmatrix} \quad (7.3)$$

where β_i is defined relative to a reference value.

In general, therefore, $\begin{bmatrix} K_r \end{bmatrix}$ is a nonlinear function of β_i due to the triple product and inversion in equation (7.2). Thus, to compute $\begin{bmatrix} K_r(\beta_i) \end{bmatrix}$, the coordinate reduction represented by equation (7.2) must be repeated for each combination of values of β_i .

In most flutter optimization procedures, the derivative of the stiffness matrix, $\frac{\partial}{\partial \beta_i} \begin{bmatrix} K_r(\beta_i) \end{bmatrix}$, with respect to many design variables β_i is required. The procedure is as follows:

Equation (7.2) is equivalent to

$$\begin{bmatrix} K_r \end{bmatrix} = \begin{bmatrix} GR \end{bmatrix}^T \begin{bmatrix} K_b \end{bmatrix} \begin{bmatrix} GR \end{bmatrix} \quad (7.4)$$

where

$$[GR] = \begin{bmatrix} -\frac{1}{[K_{22}]} & -[K_{21}] \end{bmatrix} \quad (7.5)$$

The derivative of the reduced stiffness matrix with respect to any design variable β_i , evaluated at a given combination of values of β_i , is then defined by:

$$\frac{\partial}{\partial \beta_i} [K_r(\beta_i)] = [GR(\beta_i)]^T [\Delta K_i] [GR(\beta_i)] \quad (7.6)$$

where ΔK_i is consistent with equation (7.3).

The application of equation (7.6) is as follows. The incremental stiffness matrices $[\Delta K_i]$ are invariant during the optimization process. As a new set of design variables, β_i , is defined during a step in the optimization process, $[K_b]$ is computed according to equation (7.3) and $[K_r]$ according to equation (7.2). The reduced stiffness matrix $[K_r]$ can then be used in a vibration analysis. The associated coordinate reduction matrix $[GR]$ is formed and used in the triple product of equation (7.6) to compute the derivatives of $[K_r]$ with respect to all design variables.

When the number of structural coordinates is not too high, the coordinates to be eliminated can be restricted to those that are of no interest to aeroelastic analyses and, in fact, can be considered unloaded. However, when the structural model is designed for stress analysis it may be desirable to eliminate coordinates that are of interest to aeroelastic analyses, such as deflections perpendicular to lifting surfaces. This usually means that coordinates must be eliminated that have associated inertia. If that is the case, equations (7.4) and (7.6) must be applied to the mass matrix as well (Reference 31).

In using a coarse grid finite element model for aeroelastic analysis, the aim is to reduce the number of coordinates to be eliminated to a minimum. In a finite difference approach, as described in Reference 32, the only structural degrees of freedom are deflections perpendicular to the lifting surface.

Which approach will be favored in future optimization work is hard to foresee. There seem to be three areas of investigation that could lead to significant development.

It seems most advisable, because of the directness of the approach, to speed up the computation of $[K_r(\beta_i)]$ for arbitrary sets of β_i in the basic finite element analysis system. Possibly approximate methods can be developed, which are valid for a few resizing steps, after which an exact updating takes place. It seems self-evident that the last updating in an optimization should be exact.

A second area of investigation could be based on an approach used with some success at the Lockheed-California Company. In it the reduced stiffness matrix is approximated by a polynomial function of the design variables:

$$[K_r(\beta_i)] \cong [K_r(0)] + \sum \beta_i [A_i] + \sum \sum \beta_i \beta_j [B_{ij}] \quad (7.7)$$

where the summation is over $i = 1 \rightarrow n$ and $j = 1 \rightarrow n$.

Such a polynomial can be an acceptable approximation over limited ranges of the values of the design variables. Since the stiffness is represented as an explicit function of the design variables, it can readily be evaluated for any arbitrary combination of values of β_i . The derivative of the stiffness matrix is:

$$\frac{\partial}{\partial \beta_i} [K_r(\beta_i)] = [A_i] + \sum \beta_j [B_{ij}] \quad (7.8)$$

One element of $[K_r(\beta_i)]$ is approximated by

$$K_r(\beta_i) \cong K_r(0) + \sum \beta_i A_i + \sum \beta_i \beta_j B_{ij} \quad (7.9)$$

To determine the values of the coefficients A_i and B_{ij} , $K_r(\beta_i)$ must be computed for $n+n^2$ values of β_i . Thus, to define the polynomial expression in equation (7.7) it is necessary to compute $[K_r(\beta_i)]$ for $1+n+n^2$ linearly independent columns $\{\beta_i\}$, with the help of equations (7.3) and (7.2).

On an arrow wing, where design variables were torsional and bending stiffness over certain areas of the outer wing, it was found there was little coupling between the design variables. When that is the case equation (7.7) can be written as:

$$[K_r(\beta_i)] = [K_r(0)] + \sum (\beta_i [A_i] + \beta_i^2 [C_i]) \quad (7.10)$$

Only $1+2n$ evaluations of $[K_r(\beta_i)]$ are necessary to compute $[K_r(0)]$ and all the coefficient matrices $[A_i]$ and $[C_i]$ in equation (7.10).

A third area of investigation is the use of the aeroelastic model. In that case it seems mandatory that a direct two-way relationship be developed between the sizing in the stress model and the sizing in the aeroelastic model.

7.2 Multiple Flutter Speed Constraints

Although the problem of multiple flutter speed constraints is addressed in the literature (e.g., Reference 33), examples in the literature, used to illustrate methods of optimization for flutter, are all restricted to one flutter speed. Often it is indeed possible to eliminate all flutter constraint violations by eliminating the most critical flutter speed. In general, however, the possibility of more than one active flutter speed constraint must be anticipated.

Formally, the penalty function method (Section 6.4) and the method of feasible directions (Section 6.5) have built-in capability to handle multiple flutter speed constraints. The other methods discussed in Section 6 require added logic to handle multiple flutter speed constraints.

Reference 34 makes use of the multi-constraint capability of the penalty function method by requiring that the flutter roots for selected values of the reduced frequency, k , correspond to combinations of speed and damping that provide adequate damping within the flight envelope (see also Section 7.3).

The optimality criterion for one flutter speed constraint is:

$$\frac{\partial V}{\partial m_i} = \frac{\partial V}{\partial m_j} \quad (7.11)$$

where i and j refer to free design variables, i.e., design variables that are not at a sizing constraint (References 29 and 1).

For two flutter speed constraints the optimality criterion is (Reference 1):

$$\begin{vmatrix} 1 & 1 & 1 \\ \frac{\partial V_1}{\partial m_1} & \frac{\partial V_1}{\partial m_2} & \frac{\partial V_1}{\partial m_3} \\ \frac{\partial V_2}{\partial m_1} & \frac{\partial V_2}{\partial m_2} & \frac{\partial V_2}{\partial m_3} \end{vmatrix} = 0 \quad (7.12)$$

Equation (7.12) must be satisfied for any combination of three free design variables. It is satisfied if:

$$\frac{\partial V_1}{\partial m_i} + \frac{\partial V_2}{\partial m_i} = \frac{\partial V_1}{\partial m_j} + \frac{\partial V_2}{\partial m_j} \quad (7.13)$$

Extension of this criterion to more than two flutter speed constraints is straightforward.

It was found that the value $\frac{\partial V_1}{\partial m_i} + \frac{\partial V_2}{\partial m_i}$ determines the resizing column that is generated by the feasible direction method of Reference 17 (Section 6.5) with a push-off factor $\theta = 1$. It is believed that this value can also be used in defining a resizing column if the optimization is based on the methods discussed in Sections 6.2, 6.3 and 6.6. This is further discussed in Reference 1.

To demonstrate multiple flutter speed constraint capability, the numerical examples must relate to a realistic design environment in which two or more in-flight modes lead to flutter speeds below the minimum required flutter speed. These in-flight modes may be unrelated flutter modes for a particular weight configuration of the airplane and one particular Mach number, or they may be related or unrelated flutter modes for more than one weight configuration and Mach number.

7.3 Damping Constraints

In the discussions in Section 6, the emphasis is on flutter speed constraints. This is in recognition of the flutter speed margins as defined by the Federal Airworthiness Regulations and military specifications. The requirements of the Federal Airworthiness Regulations are illustrated in

Figure 7-1. The airplane shall be designed to be flutter free within the altitude-speed envelope defined by M_{30} , $1.2 M_D$, $1.2 V_D$ and $h = -3100$ m (-10,200 ft). In addition to this flutter speed requirement there is an implied requirement for adequate modal damping within this envelope. This is illustrated in Figure 7-2: for all flight conditions within the flight envelope there must be a certain amount of positive damping. In k-method terminology this means $g \leq g_{\max}$, and in p-k-method terminology $\gamma \leq \gamma_{\max}$. Both g_{\max} and γ_{\max} are negative quantities. From the definitions of g and γ it follows that for small values $g \cong 2\gamma$.

To satisfy the general damping constraint, the inequality constraint $\gamma \leq \gamma_{\max}$ (or $g \leq g_{\max}$) must be invoked at several speeds below $1.2 V_D$, for all in-flight modes of interest, for several Mach numbers and for the airplane weight configurations to be considered. In a realistic design environment this may lead to hundreds of inequality constraints. It is obvious there are practical difficulties associated with that many constraints.

Experience shows that usually only very few damping constraints are active and they are associated with hump modes. Sometimes damping constraint violations by hump modes disappear as the structure is resized to eliminate the most critical flutter speed violation(s). In anticipation of the need for invoking a minimum hump mode damping constraint, Section 3.4 presents a procedure to directly determine the minimum damping of a hump mode.

The optimality criterion for an active damping constraint is:

$$\frac{\partial \gamma}{\partial m_i} = \frac{\partial \gamma}{\partial m_j} \quad (7.14)$$

where i and j refer to free design variables. It corresponds to the optimality criterion for a flutter speed constraint, the derivation of which (given in Reference 1) can be generalized to arbitrary constraints.

For a combined flutter speed and damping constraint, the optimality criterion is:

$$\begin{vmatrix} 1 & 1 & 1 \\ \frac{\partial V}{\partial m_1} & \frac{\partial V}{\partial m_2} & \frac{\partial V}{\partial m_3} \\ \frac{\partial \gamma}{\gamma_{m_1}} & \frac{\partial \gamma}{\gamma_{m_2}} & \frac{\partial \gamma}{\gamma_{m_3}} \end{vmatrix} = 0 \quad (7.15)$$

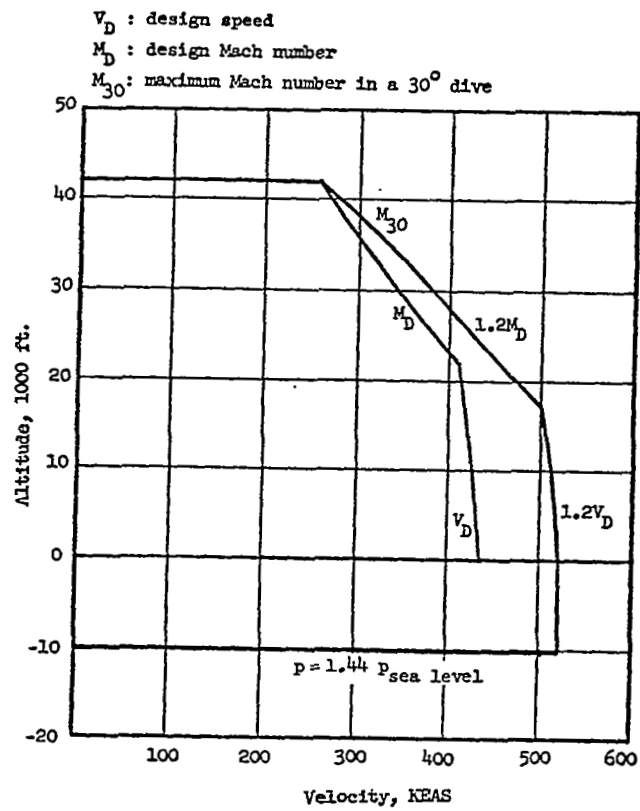


Figure 7-1: Example of Flight Envelope

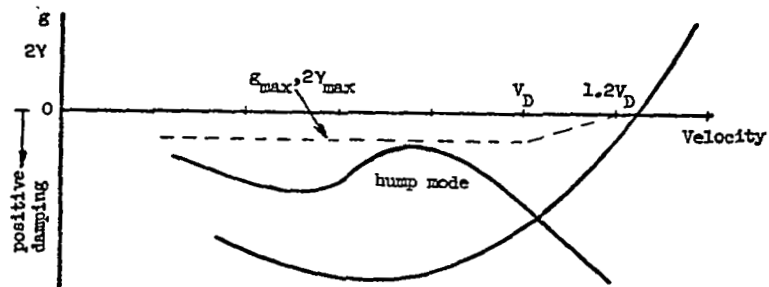


Figure 7-2: Minimum Damping Requirement

Equation (7.15) must be satisfied for any combination of three free design variables. It is satisfied if

$$\frac{\partial V}{\partial m_i} + C \frac{\partial \gamma}{\partial m_i} = \frac{\partial V}{\partial m_j} + C \frac{\partial \gamma}{\partial m_j} \quad (7.16)$$

Here C is an arbitrary constant that can be used to create compatible units or to assign a different weighting to the two constraints.

In several methods of optimization this criterion can provide a guide towards generating a resizing column.

Further development of a practical method of including damping constraints in flutter optimization should be based on numerical examples in a realistic design environment.

7.4 Mass Ballast

The literature pays little or no explicit attention to ballast (dead weight) as a design variable. The reason for this omission is understandable: any method of optimization that can handle design variables representing related stiffness and mass changes can handle a design variable representing a mass change only.

Adding mass ballast may be a more efficient way of raising the flutter speed to its required value than structural stiffening. That appeared to be the case on one of the United States supersonic transport designs and in the example treated in Reference 29. In the latter case, however, it is not clear whether the modified strength requirements due to the addition of ballast are accounted for. Reference 29 demonstrates a potential complication associated with mass ballast as a design variable: as ballast is added in a particular region, the flutter speed derivative with respect to the mass ballast changes from negative to positive. As stated in Reference 29, if this phenomenon occurs, an automated resizing procedure may fail to recognize the beneficial effect of a larger amount of ballast since an infinitesimal amount of ballast proved to lower the flutter speed. Until this aspect of flutter optimization has received more attention, considerable engineering judgment should be used in handling mass ballast as design variables.

In view of the preceding paragraph, it may be convenient to first consider stiffness design variables and their associated masses only for raising the most critical flutter speed to the desired value and the subsequent optimization at constant flutter speed. Starting with the original deficient configuration, it is then determined whether any mass change without stiffness change is more efficient than the most efficient stiffness change in raising the flutter speed to the desired value. Incremented Flutter Analysis can be

used to directly determine the amount of ballast needed to meet the flutter constraint, regardless of any changes in sign of the flutter speed derivatives as a function of the amount of mass ballast. If mass ballast is more effective than optimum stiffening, the mass ballast design variables of interest can be added as design variables for a final optimization process.

7.5 Interface With Strength Optimization

Combined optimization for flutter and stress has been demonstrated with simple structural models and/or under simplifying assumptions (References 9, 16 and 29).

In References 9 and 16 the penalty function method is used and, in order to reduce the number of stress constraint derivatives to be evaluated, the stress constraint is reduced to one constraint per loading condition. The effect of this can be easily seen in the following formulation of the penalty function approach.

The modified objective function may be represented by:

$$\phi(m_i) = \sum_{i=1}^n m_i + \frac{r_V}{V(m_i) - V_R} + \sum_{j=1}^n \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i)} + \sum_{i=1}^n \frac{r_m}{m_i - \bar{m}_i} \quad (7.17)$$

where:

$V(m_i)$ = flutter speed

V_R = minimum desired flutter speed

$\sigma_j(m_i)$ = stress in j th element $j = 1 \dots n$.

$\bar{\sigma}_j$ = maximum allowable stress in j th element

m_i = design variable; mass associated with i th design element;
 $i = 1 \dots n$.

\bar{m}_i = minimum allowable value of m_i

r_V, r_σ, r_m = penalty weighting factors

It is understood that in equation (7.17) there are as many $\frac{r_V}{V(m_i) - V_R}$ terms as there are flutter speed constraints, and as many $\sum_{j=1}^n \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i)}$

terms as there are design load conditions. For this discussion it is sufficient to assume one flutter speed constraint and one design load condition.

As a part of the determination of the resizing column, the partial derivatives $\partial\phi(m_i)/\partial m_i$ are used.

$$\frac{\partial\phi}{\partial m_i} = 1 - \frac{r_V}{(V(m_i) - V_R)^2} \frac{\partial V}{\partial m_i} + \frac{\partial}{\partial m_i} \sum_{j=1}^n \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i)} - \frac{r_m}{(m_i - \bar{m}_i)^2} \quad (7.18)$$

For each design variable, one derivative $\frac{\partial V}{\partial m_i}$ and n derivatives $\frac{\partial}{\partial m_i} \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i)}$ must be evaluated. Thus a total of n flutter speed derivatives and n^2 stress derivatives must be evaluated. In References 9 and 16 the number of stress derivatives is reduced to n by evaluating the derivatives by means of finite differences but performing the differentiation after the summation; i.e., the following identity

$$\frac{\partial}{\partial m_i} \sum_{j=1}^n \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i)} = \sum_{j=1}^n \frac{r_\sigma}{(\bar{\sigma}_j - \sigma_j(m_i))^2} \frac{\partial \sigma_j}{\partial m_i} \quad (7.19)$$

which requires the evaluation of n derivatives per single design variable, is replaced by the finite difference representation:

$$\frac{\delta}{\delta m_i} \sum_{j=1}^n \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i)} = \frac{1}{\delta m_i} \left\{ \sum_{j=1}^n \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i + \delta m_i)} - \sum_{j=1}^n \frac{r_\sigma}{\bar{\sigma}_j - \sigma_j(m_i)} \right\} \quad (7.20)$$

requiring the evaluation of only one derivative per single design variable. This substitution does not affect the one-dimensional minimization that is part of the method used in Reference 14, but it does affect the direction of the resizing vector. Instead of each elemental stress contributing

individually to $\frac{\partial\phi}{\partial m_i}$, one contribution representing an average stress

penalty term is used. No studies investigating the effect of this substitution have been reported.

In Reference 28, the optimality criterion $\frac{\partial V}{\partial m_i} = \text{constant for all } i$ is used in the flutter optimization and the fully-stressed-design criterion for strength optimization. The two optimizations are performed alternately until a converged design is obtained.

The fully-stressed-design criterion does not, in general, lead to a minimum weight structure, at least not for a redundant structure (Reference 35). Furthermore, without further investigation there is little ground for expecting that alternating flutter and strength optimization will lead to a converged design if both flutter constraints and stress constraints are active. This leads to the conclusion that ideally flutter and strength optimization should take place simultaneously as is done in References 9 and 16.

It would seem that the adequacy of the methods of References 9, 16 and 29 has not been demonstrated when applied to a practical design problem. Conceptually the methods of References 9 and 16 allow as many stress constraints per load condition as there are independent stress constraints (which may be larger than the number of elements). Thus the number of independent stress constraints can be very large. This has led to the definition of one stress constraint per load condition, which, however, removes the possibility of independent stress constraints contributing individually to the move vector direction. Possibly other composite stress constraints can be defined such that the number of derivatives to be evaluated is reduced while retaining the contribution of each constraint to the move vector direction.

Additional investigations in the areas of structural modeling and combined optimization for flutter and strength are needed before conclusions regarding the best approach can be formulated. Such conclusions should be based on the results of analyses in a realistic design environment.

8. COMPUTATIONAL ASPECTS OF THE FLUTTER TASK

The purpose of this section is to delineate the computational aspects of the complete flutter task, which includes flutter analysis as well as structural synthesis, i.e. the design of a structure that satisfies the flutter requirements. Although the subject of this report is flutter optimization, i.e. structural synthesis aimed at a minimum weight structure that satisfies the flutter requirements, it is useful to include flutter analysis, or flutter survey, in this discussion, since there is a large common data base and many common analytical tools.

Structural design aimed at satisfying flutter requirements must, to result in a viable airplane, also take into account strength requirements and requirements related to manufacturing cost. Examination of a merit function that

combines structural weight and manufacturing cost falls outside the scope of this study. The interface between structural synthesis with strength constraints and synthesis with flutter constraints is discussed in Section 7.5. There it is concluded that additional investigations in the areas of structural modeling and structural optimization with combined flutter and strength constraints are needed before the best approach to total structural synthesis can be formulated. The organization of computer programs and modules discussed in this section as a possible basis for definition of specifications for computer software envisions combining flutter optimization with satisfaction of stress constraints.

8.1 The Complete Flutter Task

The complete flutter task can be considered as being composed of three subtasks, subsequently to be discussed:

1. Flutter survey of the original design.
2. Initial structural resizing to satisfy all flutter requirements.
3. Flutter optimization: weight minimization while explicitly satisfying flutter requirements and not violating strength requirements.

8.1.1 Flutter Survey - The original design is defined by the external geometry, by a structural mass distribution derived by satisfying strength requirements, by an additional mass distribution representing fixed, non-structural airframe masses (e.g. powerplants, control system, furnishings) and mass distributions representing useful loads (e.g. fuel, payload).

The flutter survey is a series of flutter analyses sufficient to identify any flutter deficiency that may exist in the original design over a range of operating conditions. Although details of the actual execution of the flutter survey may be different for different engineering facilities, it is believed that the flow diagram in Figure 8-1 is generally applicable. The general procedure is not new; it is discussed here in order to relate it to the overall design process. In Figure 8-1 oval boxes define engineer action points, although not necessarily manual operations; rectangular boxes represent computing modules. The computing process can proceed from one module to the next without engineer action, although an engineer's review may be inserted at any point.

Starting with the design definition box in Figure 8-1, the engineer proceeds to prepare, not necessarily manually, structural model data, inertia data and aerodynamics data. The structures module forms the stiffness and inertia matrices that are used in the vibration analysis. It performs, if necessary, the static coordinate reduction to keep the number of degrees of

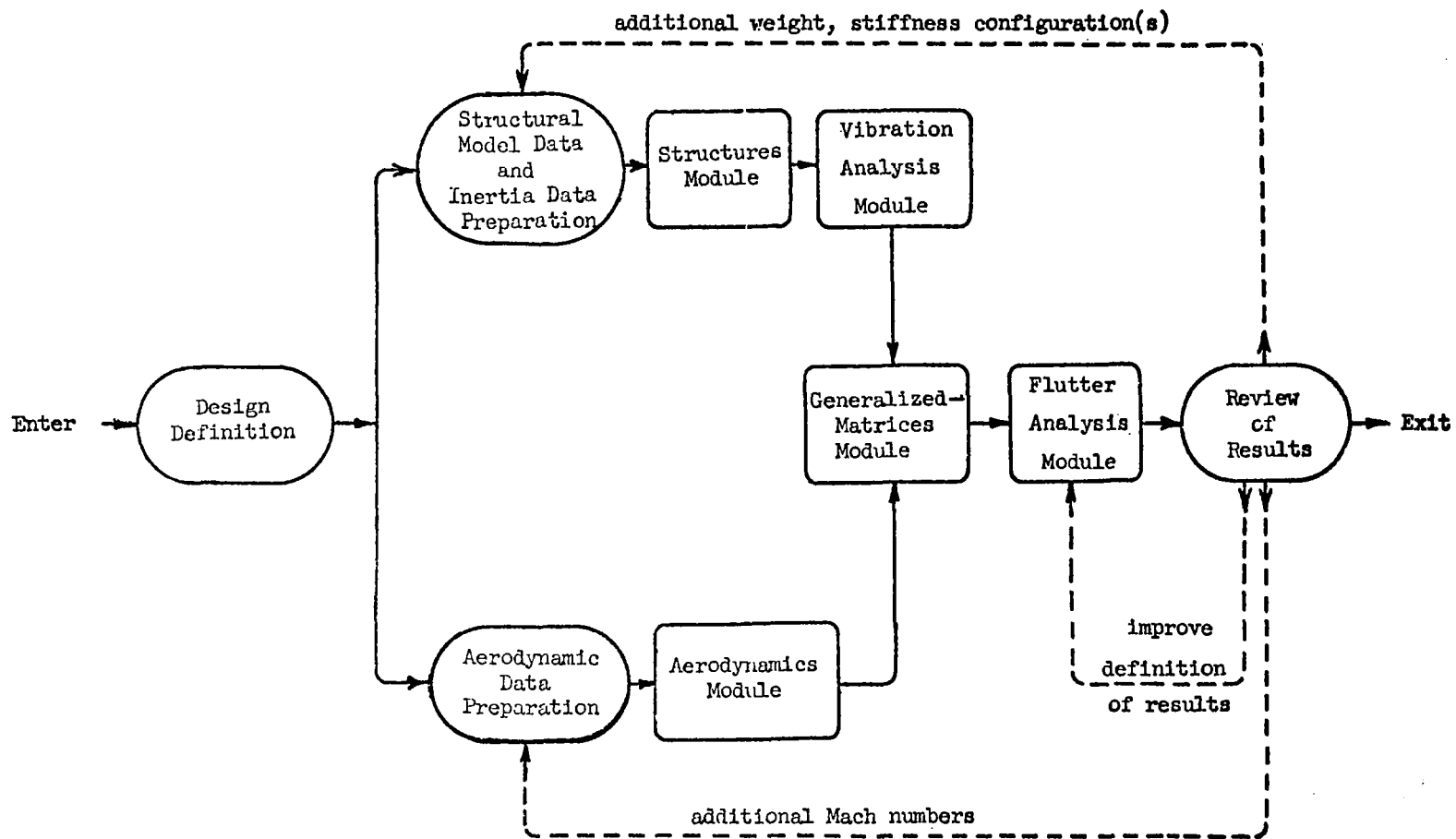


Figure 8-1: Flutter Survey Task

freedom for the vibration analysis within a practical limit. The structures module may also form the inertia matrix associated with the masses of the structural elements. If this is the case, the inertia data prepared by the engineer refer to nonstructural and useful load masses only. Inertia matrices for a number of useful load configurations, chosen on the basis of experience, and the output of the structures module are input into the vibration analysis module and vibration analyses, leading to natural frequencies and vibration modes, are performed.

The natural frequencies and vibration modes can be reviewed by the engineer for checking purposes and, after the flutter analysis, for obtaining a better understanding of the flutter behavior of the airplane. In the case of a final design, the results of the vibration analysis can be compared with the results from ground vibration tests. From an analytical point of view, however, the vibration analysis is only necessary if the original number of degrees-of-freedom exceeds a practical limit for the flutter analysis. In that case, the vibration modes associated with the lower vibration frequencies are, in general, used as generalized coordinates for the flutter analysis. The output of the vibration analysis can be formulated to include generalized stiffness and inertia matrices. In that case, the function of the generalized-matrices module is to form only the generalized aerodynamics matrices. However, to be more generally useful, this module should also include the capability of generating generalized mass and stiffness matrices by pre- and postmultiplication by modal matrices. This capability may be used if the generalized coordinates are not updated after each resizing step. In view of the options available for forming the generalized aerodynamics matrices (Section 5.3), the generalized-matrices module may do more than a pre- and postmultiplication by the modal matrices obtained from the vibration analysis. Consequently, the output of the aerodynamics module may be a set of $[HAW]$ matrices (Equation (5.16)) for discrete values of the reduced frequency k , or may be a set of basic aerodynamic influence coefficient matrices $[AIC(k)]$ and matrices $[H]$, $[DX]$ and $[DZ]$ (Equations (5.8) and (5.32)), or any combination between these extremes as discussed in Section 5.4.4. The aerodynamics data preparation consists of defining aerodynamics grid systems for the downwash collocation points and the aerodynamics loads points, and the selection of Mach numbers for which the flutter analysis will be performed. Structural grid data are input into the aerodynamics module to form the grid transformation matrices $[H]$, $[DX]$ and $[DZ]$.

Static reduction of the stiffness matrix is a generally accepted method of reducing the number of degrees of freedom in the vibration analysis. Reference 36 presents an alternative that is worth consideration. In the approach of Reference 36, no reduction of the stiffness matrix takes place; in fact, the complete stiffness matrix is not assembled. Instead, the output of the structures module is a collection of submatrices that are used directly in the vibration analysis module to compute generalized stiffness and inertia data, and vibration modes.

From the generalized-matrices module the flutter survey procedure enters the flutter analysis module. The flutter analysis module should contain an interpolation routine for computing the generalized aerodynamics matrix for arbitrary k values. The flutter equation is solved by any suitable method and the output is a series of f - g - V diagrams (Figure 3-1) for different Mach numbers, different distributions of useful load and for symmetric, anti-symmetric and possibly asymmetric modes.

Figure 8-1 indicates three potential reanalysis loops. Any realistic procedure must account for the possibility that the initial choice of input parameters does not provide sufficient definition of the flutter characteristics of the original design. It is possible that the initial choice of k -values or speeds for which the flutter analysis is performed is insufficient to determine flutter speeds or minimum damping in hump modes. Thus, review of the flutter analysis results may require return to the flutter analysis module for improved definition of the f - g - V diagrams. Review of the results may also indicate the need for including more Mach numbers, inertia configurations associated with different useful load distributions, or stiffness matrices in the flutter survey. The possibility of including more than one stiffness matrix follows from the fact that failed conditions must be considered.

After sufficient flutter data have been generated, it is determined whether any flutter deficiencies exist. If there are no deficiencies, the flutter task is completed, unless design changes occur which make it necessary to repeat the flutter survey. Since the first survey resulted in engineering familiarity with the flutter characteristics of the design, additional surveys usually can be restricted to fewer combinations of inertia configurations, Mach numbers and failed conditions than were investigated during the first survey.

If there are flutter deficiencies, a structural resizing is initiated which is aimed at satisfying all flutter requirements.

8.1.2 Initial Structural Resizing - If the flutter survey of the original design indicates the presence of flutter deficiencies, it is desired to remove these deficiencies with a minimum weight penalty. Reference 1 indicates that it is theoretically possible to attain a minimum weight design by judiciously adding structural mass in small quantities to those structural elements that, at each step, are most efficient in removing the deficiencies. For a structure with a large number of design variables and several flutter deficiencies, this is, however, an impracticable approach. It is more efficient to first generate, in one or very few resizing steps, a structure without flutter deficiencies, but not necessarily with minimum weight, and then minimize the weight while avoiding flutter deficiencies. Most methods of flutter optimization discussed in this report are based on this approach.

Two types of flutter deficiencies are recognized: 1) too low a flutter speed ($V_f < V_R$) and 2) insufficient damping at the top of a hump mode

($\gamma_{\text{hump top}} > \gamma_{\text{max allowed}}$). Although both deficiencies are undesirable, the flutter speed deficiency occurs more frequently and is emphasized throughout this report. Optimization techniques aimed at satisfying flutter speed requirements can be generalized to include damping requirements. In this discussion, reference will be made, for convenience, to flutter speed requirements, or constraints, only.

Experience at the Lockheed-California Company, related to a realistic design environment, suggests that on the basis of engineering judgment, one flutter deficiency often can be identified as being most critical. That is, for a particular Mach number and useful load configuration there exists a deficiency, the removal of which is expected to result in all flutter deficiencies being removed. If this is not the case, then one or more additional applications of the following approach will lead to a design without flutter deficiencies. Neither case, in general, leads to an optimum design.

Reference (1) indicates that a resizing column

$$\left\{ \Delta m_i \right\} = C \left\{ \frac{\partial V_{mc}}{\partial m_i} \right\} \quad (8.1)$$

where V_{mc} is the most critical flutter speed, and C defines a magnitude such that $V_{mc} = V_R$, is an efficient initial resizing. The column $\left\{ \frac{\partial V_{mc}}{\partial m_i} \right\}$ is recognized as the gradient of the flutter speed. Other distributions of Δm_i , however, may be considered, e.g.,

$$\left\{ \Delta m_i \right\} = C \cdot \text{positive elements of} \left(\left\{ \frac{\partial V_{mc}}{\partial m_i} \right\} - \left\{ \frac{1}{\partial V_{mc} / \partial m_i} \right\} \right) \quad (8.2)$$

If it is difficult to define one most critical flutter mode, a resizing column based on a weighted sum of two or more flutter speed gradients may be a better approach. In any case, initial resizing should take into account the efficiency with which design variables can increase the flutter speed and, thus, it is necessary, at this stage, to define design variables and to determine partial derivatives of the flutter speeds with respect to the design variables. In general, the initial resizing column can be defined as:

$$\left\{ \Delta m_i \right\} = C \left\{ f \left(\frac{\partial V_j}{\partial m_i} \right) \right\} \quad (8.3)$$

where the subscript j refers to the flutter speeds that are less than the required speed.

The initial resizing procedure, again, may be different for different engineering facilities and it probably depends, in its details, on the preceding flutter survey procedure as well as on the subsequent optimization procedures to be used. With this in mind the essential features of the procedure are shown in the flow diagram of Figure 8-2. Note that in Figure 8-2 oval boxes still indicate engineer action points, but the rectangular boxes no longer define computational modules but computing activity in general, and no attempt is made to define specific modules.

The endpoint of Figure 8-1 is the starting point for Figure 8-2: review of the flutter survey results. If there are flutter deficiencies, design variables may be defined and most critical flutter conditions selected. To

generate the flutter speed derivatives $\frac{\partial V_j}{\partial m_i}$, the characteristic roots and

vectors corresponding to the flutter points must be determined (e.g., by the two-dimensional Regula Falsi followed by a subroutine for determining characteristic vectors). In addition, derivatives of the mass, stiffness and aerodynamics matrices are required. If there is a static reduction of the stiffness matrix, the static reduction matrix must be explicitly generated in order to compute the derivatives of the reduced stiffness matrix. The

flutter speed derivatives $\frac{\partial V_j}{\partial m_i}$ can be computed following the formulation of

Reference 14, a compact version of which is included in Reference 1. The

analyst may want to review the values of $\frac{\partial V_j}{\partial m_i}$ before deciding on the direc-

tion of the resizing column, $\left\{ f \left(\frac{\partial V_j}{\partial m_i} \right) \right\}$, or it may be formed automatically

by the program. Incremented Flutter Analysis (References 1 and 4), is a con-

venient method of determining each C_j that results in $V_j = V_{R_j}$. If

$\left\{ f \left(\frac{\partial V_j}{\partial m_i} \right) \right\}$ contains only positive elements, the largest value of C_j determines the resizing column $\{\Delta m_i\}$ according to equation (8.2) which results

in all $V_j \geq V_{R_j}$. If there is uncertainty about this result, the structure

is incremented by $\{\Delta m_i\}$ and a new vibration and flutter analysis is per-

formed for selected combinations of Mach number and useful load. If necessary,

more critical flutter speeds V_j are selected and the process is repeated.

The procedures described, and illustrated in Figure 8-2, are aimed at a one-step resizing to reach the goal of satisfying all flutter requirements. Variations of this procedure are possible but would include the same basic computational modules.

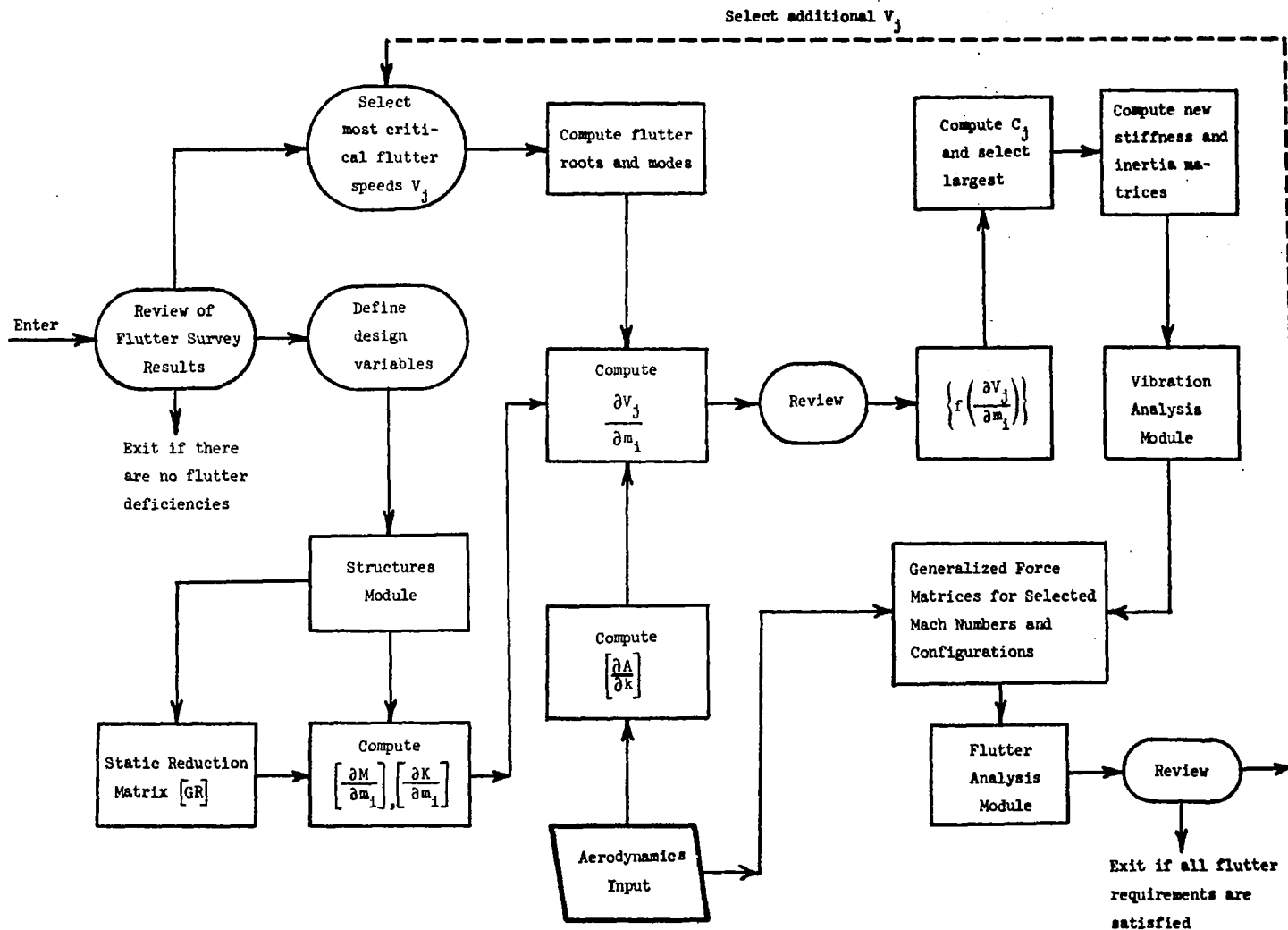


Figure 8-2: Initial Structural Resizing Task

8.1.3 Flutter Optimization - Of the three subtasks discussed in this section, the flutter optimization task is most dependent on the computational methods adopted by an engineering facility. However, this method dependency is mainly concentrated in the actual resizing procedure with a constant set of generalized coordinates (invariant vibration modes). The structural analysis and vibration analysis, to be repeated several times during the optimization task, can be defined in general terms, relatively independent of the resizing procedure used.

The methods of optimization discussed in Section 6 deal with only a small aspect of the optimization task, namely, the resizing when given a constant set of generalized coordinates (i.e., invariant vibration modes). It is most likely that modal updating is necessary (Section 4). If a static reduction of the stiffness matrix is used to reduce the number of degrees of freedom in the vibration analysis (Section 7.1), the reduced stiffness matrix may be a nonlinear function of the design variables. This results in additional computations to determine derivatives of the stiffness matrix and possibly the inertia matrix.

In this section the overall flutter optimization task, except for the resizing for a constant set of generalized coordinates, is delineated under the following assumptions:

1. There will be modal updating.
2. Static reduction of the stiffness matrix is required.
3. For each resizing step the derivative of the reduced stiffness matrix is determined exactly (Equation (7.6)).
4. No static reduction of the mass matrix is required.
5. There is one active flutter speed constraint.
6. Strength requirements are satisfied.

Keeping in mind these assumptions, a flow diagram is formulated (Figure 8-3) that delineates those computational steps that are considered independent of the method chosen for determining the resizing steps, given the flutter speed derivatives. Several of the computations indicated in Figure 8-3 are identical to those in Figure 8-2.

The inputs into the flutter optimization task are: the starting stiffness and mass matrices that are output from the initial resizing task; the incremental stiffness and mass matrices; the basic aerodynamics input; and the aerodynamics derivatives input.

The starting (first current) stiffness and mass matrices are used to define generalized coordinates via a vibration analysis. These generalized coordinates along with the associated generalized stiffness and mass matrices,



Figure 8-3: The Flutter Optimization Task

and the basic aerodynamics input are used to obtain a point solution of the flutter equation and the associated characteristic vectors. The characteristic vectors are combined in turn with the incremental stiffness matrix pre- and postmultiplied by the static reduction matrix $[GR]$, with the incremental mass matrix, and with the aerodynamics derivatives input, to form the derivative scalars (See Reference 1). The flutter speed derivatives are formed and input into the resizing module. Other inputs into the resizing module depend on the method of optimization used, but include minimum size constraints equal to the original design, i.e., before the initial resizing, and program control parameters to be chosen by the analyst.

The resizing module generates a column of design variable increments defining one step in a resizing process that usually comprises several steps. The total mass associated with the new values of the design variables is compared with the previous total mass. If the total mass has not converged to a minimum, the design variable increments are used to generate new current stiffness and mass matrices and the process is repeated.

Satisfying the stress constraints can be accomplished in various ways. One approach, which does not involve the resizing module, is shown in Figure 8-3. In it, when a minimum total mass is reached, the loads and stress analysis is redone. Due to stiffening of the original design, there may be stress violations due to redistribution of internal loads or changes in the external loads. Where such violations occur, the element sizes are increased to satisfy the stress constraints and new current stiffness and mass matrices are formed and the flutter optimization is repeated using updated minimum sizes. After a new convergence on a minimum total mass, the stresses are checked again and, if necessary, the entire process is repeated. The increased element size will, in general, cause a change in flutter speed. This is expected to be small, since the stress violations are most likely to be in elements that have not increased in size during the flutter optimization and, thus, are ineffective in changing the flutter speed.

In another approach there is a strong interaction between a stress analysis module and the resizing module such that each resizing step is constrained such that stress constraints are not violated. Two possibilities can be distinguished: the loads are assumed constant or the loads are recalculated at each step. In the former case the loads and stress analysis shown in Figure 8-3 must follow the flutter optimization as described in the previous paragraph.

It is noted that the aerodynamics input, in Figure 8-3, is not defined in terms of specific matrices. The discussions in Section 5 and Reference 1 indicate that there are many options for formulating the matrices of generalized aerodynamic force coefficients, and it is considered outside the scope of this discussion to present a definite choice. It is worthwhile noting, however, that if cubic spline interpolation is used for the aerodynamics, the basic aerodynamics input and the aerodynamics derivatives input are identical matrices.

Since no recommendations for a specific method of optimization are made as a result of this study, a further definition of the resizing module, in Figure 8-3, is considered to fall outside the scope of this report.

8.2 Aspects of the Computing System

General aspects of the computing system required to perform the flutter task are discussed without attempting to define detailed specifications for such a system.

When properly divided into well defined building blocks the complete flutter task, including the optimization, is relatively straightforward, inviting extensive automation. The flutter optimization task seems especially well suited for complete automation. It is questionable, however, whether the actual design experience available is sufficient to decide on all aspects of the optimization task and to embark on the design of a computing system that can handle efficiently a structural design that is defined by several thousands of finite elements. In view of this the Lockheed-California Company has first developed a semi-automatic system, based on its Computer Graphics system. It has been used on the configuration upon which the numerical examples of Appendix A are based, on an arrow wing supersonic transport study (Reference 1) and on an actual hardware problem. The authors believe, however, that a batch process system with maximum automation options is a desirable design asset, even if in its initial version it is somewhat restricted in the number of structural degrees of freedom, design variables and flutter constraints it can handle. In this section some aspects of a batch process computing system are discussed.

An engineering facility that already has a computing system for flutter analysis (flutter survey task) may want to restrict its batch process system to the initial resizing task and the flutter optimization task. This, however, seems only justified if the data input and output of the existing flutter analysis system can easily be made compatible with the input requirements for the other tasks. Since the repetitive analyses associated with flutter optimization put extra emphasis on computational efficiency, a facility may decide to update its flutter analysis system as part of the introduction of a flutter optimization capability. In the following it is assumed that a batch process system for the complete flutter task is to be designed.

When comparing the flow diagrams in the Figures 8-1 through 8-3 it is clear that the three tasks represented in these three figures have a large number of computational functions in common. Thus, the first point of consideration is whether three independent programs should be developed within an existing computing system or whether one program, or a new system should be developed for the complete flutter task. The choice depends on what computing system is available at a facility and, to a certain extent, personal preferences of the engineers. From a practical engineering point of view it seems self-evident that, whatever course is chosen, data format compatibility is mandatory and that a data management system is required.

In selecting a computing system for the complete flutter task, consideration should be given to a system that has access to an existing matrix algebra computing system. If this access is not available, the complete flutter task system should include some generalized matrix algebra capability to enable the user to depart from a rigid format.

In designing a computer system for the complete flutter task, two approaches can be distinguished: Self-Contained Programs and Variable Job Stepping.

In the first approach, which is the more common of the two, there may be a self-contained program for each of the subtasks comprising the complete flutter task, or some or all subtasks may be combined into one self-contained program. Each program has its own executive module which controls calling into core the various computing modules (sub-programs) as they are needed during the entire computer operation for that program. Organizing the complete flutter task in one program with one executive module would result in a very large program with some complex input/output interface problems as well as core overlay problems. If existing batch programs are to be integrated into such an overall computing program, extensive modification might be needed in order to resolve some of these problems. If the complete flutter task is covered by more than one program, automatic transfer from one program to another might prove impracticable, thus limiting the overall automation attainable. Since engineer reviews are required during the computational effort, however, this may not be an important limitation. Consideration must be given to the degree of commonality between modules in the separate programs that perform the same function. Failing to achieve complete commonality, e.g., due to different overlay requirements, increases the effort needed to update a functional module.

Variable Job Stepping consists of a number of separate computer programs each representing a computing module, such as those defined in Figure 8-1, which are controlled by another program called the Executive. Variable Job Stepping, therefore, is a sequence of separate computer job steps in which each job step is a function of the preceding job step(s) as determined by the Executive Module. The Executive Module (a separate program) monitors the task completion codes of the preceding steps and, based on the instruction code supplied by the engineer, determines which computing module is next required. Prior to transferring the control to the computing module, the Executive Module prepares the input data for that computing module in accordance with its data format requirements. The Variable Job Stepping system is inherently modular in approach. To add another module, only the Executive Module (program) needs to be modified and reloaded as a new executable program in addition to the new computing module. Existing batch process programs can be included in a Variable Job Stepping system with little modification to the batch programs. If a new method for computing, say, aerodynamic matrices, becomes available, again only the Executive Module needs to be modified. It is also worth noting that each program within the Variable Job Stepping system could be executed as an independent program outside the Executive Module Control. Figure 8-4 illustrates the principle. The Executive is loaded and

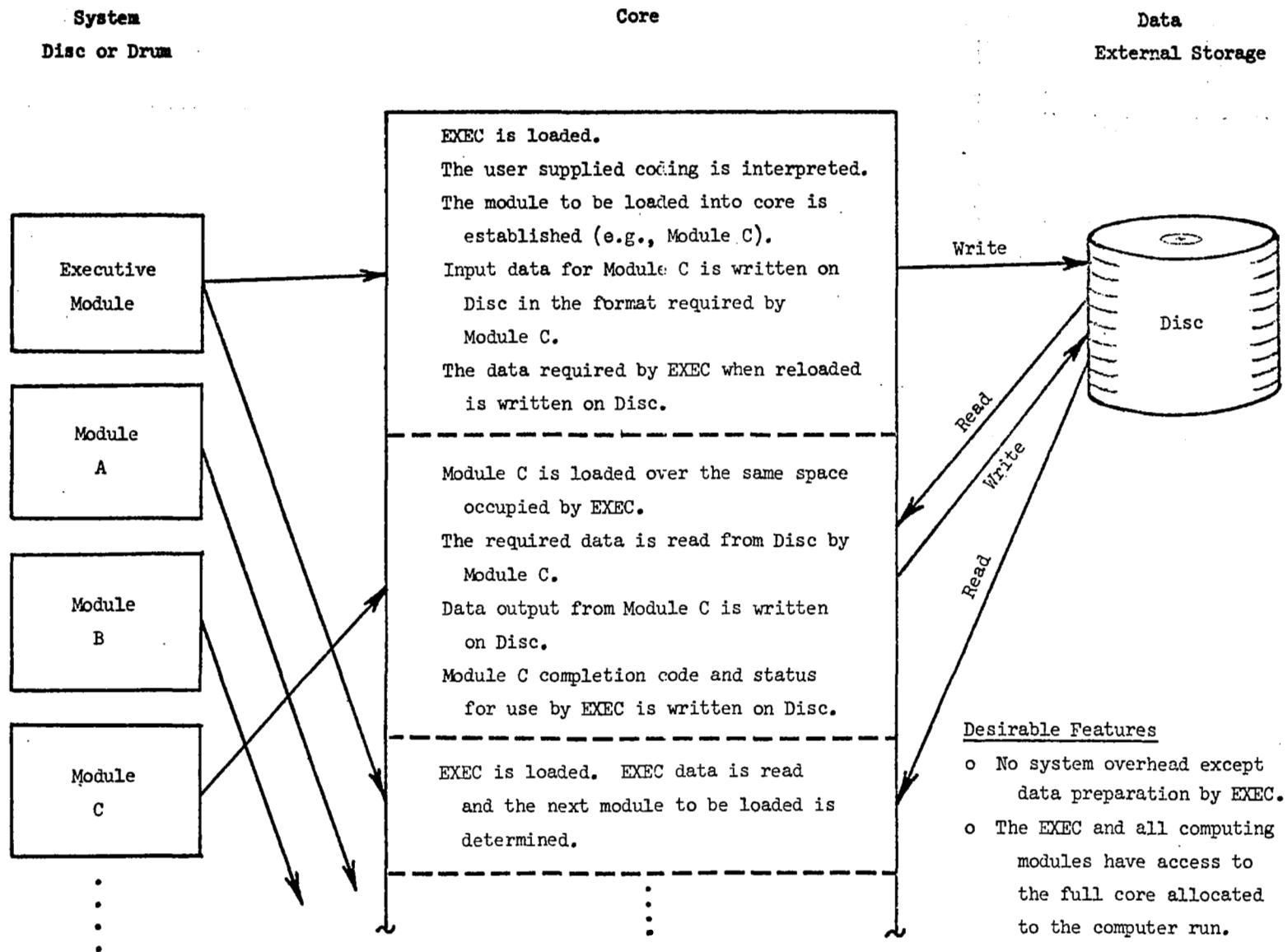


Figure 8-4: Computing System Using Variable Job Stepping

calls in the next module needed, say C, which is loaded over the same space as was occupied by the Executive Module. When module C has completed its task it calls back the Executive Module, at the same time giving it instructions that depend on the output of module C. The Executive Module calls in the next computational module, again while annihilating itself from core.

Which of the two approaches discussed is preferred depends on many factors and, without a more in-depth look at the details of the computing system, cannot be determined at this time.

As stated above, the complete flutter task is relatively straightforward, when properly defined, and will, in principle at least, not give rise to great programming difficulties. The great challenge in designing the system lies in maximizing its capacity, in terms of number of structural coordinates, design variables and flutter constraints, while keeping computing cost reasonable. This does require a very efficient use of all computer resources: maximum utilization of available core, efficient input/output routines, systematic labelling and external storage of data blocks, efficient compacting of data blocks (e.g. sparse matrices), etc. Obviously, close cooperation between experts in flutter analysis, optimization procedures and computer programming is required in order to obtain an efficient computing system.

9. CONCLUSIONS AND RECOMMENDATIONS

Based on the investigations and evaluations performed during this study and augmented by the additional supporting activities conducted concurrently at the Lockheed-California Company, it is concluded that an efficient, practicable flutter optimization module can be formulated and implemented with a reasonable amount of further development. Some of the particular conclusions supporting this general conclusion are presented in the following:

Aerodynamics parameters may be efficiently represented by the five-matrix product shown in equation (5.12). The most efficient method of implementation of this equation depends upon a complex relationship involving the numbers of aerodynamic integration points, downwash points, generalized modal coordinates, discrete structural coordinates and reduced frequencies used, as well as the interpolation procedures employed. To be completely general, several options should be available in order to provide alternate procedures for the generation of these parameters. In relation to the complete flutter optimization task, however, it is concluded that severely limiting the number of such options would not significantly increase the required computing resources.

Repetitive flutter solutions of the type required in structural resizing procedures may be efficiently obtained using the 2-D Regula Falsi method. Although other flutter solution procedures might be developed to the same degree of efficiency and reliability demonstrated by the Regula Falsi in

obtaining repetitive flutter solutions, it is concluded that this procedure has a wider range of application than other procedures considered. The 2-D Regula Falsi method can be used in the solution of a general form of the flutter equation in order to obtain the values of any two dependent variables required to satisfy the equation. It can be used in a direct form of Incremented Flutter Analysis to solve for the magnitude of the increment of a specified design variable, along with an additional variable, necessary to satisfy a flutter constraint. Other applications, such as the determination of the minimum damping of a hump mode, are also possible.

Resizing procedures vary widely in particular detail, with each procedure considered exhibiting one or more advantageous features. In terms of a realistic design effort, however, it is concluded that the general differences between the arbitrary step-size procedures and the defined step-size procedures are more significant than the differences between the individual procedures in each category. Based on the numerical evaluations of the idealized test case of Appendix A, it would appear that the arbitrary step-size procedures produce the same mass reduction as the defined step size procedures with less total computing cost. It is not clear that this same result would be obtained for a more complex design effort. It is concluded, however, that either type of procedure can be included in a flutter optimization module with no undue difficulty.

Practical considerations in the implementation of a flutter optimization procedure can have at least as profound an effect on the performance of the flutter module as does the selection of the three major elements considered thus far. These considerations include the choice of structural model, number of structural degrees of freedom retained, method of generation of incremental stiffness matrices, number of modal coordinates used, frequency of updating vibration modes and frequency of updating flutter derivatives. No definite conclusions are available regarding these considerations, since it was not possible to conduct the required investigations within the scope of the present study.

The conclusions reached in the course of the present study, and presented above, lead to the following recommendations for development of a flutter optimization module:

1. Computer coding for the aerodynamics submodule should be based on the five-matrix product of equation (5.12). The number of options in forming the five-matrix product should be limited to those forms that are most generally useful.
2. The flutter solution submodule for the repetitive flutter solution procedure should be based on the two-dimensional Regula Falsi approach. A global solution procedure, such as the p-k method of Reference 3 or the Desmarais-Bennett method (Reference 7) should be included for the initial and final flutter survey.

3. At least two candidate resizing procedures should be evaluated in a realistic design effort such as the arrow wing flutter optimization task performed under Contract NAS-1-12288. One resizing procedure should be of the arbitrary step-size type and the other a defined step-size type. To facilitate the comparison, it would be useful to maintain as much commonality between the two methods as possible. As an example, the resizing procedure formulated in Section 6.7.4 could be used as the defined step-size procedure, and the gradient projection search of Rudisill-Bhatia (Section 6.2), modified to incorporate the move vector of Section A.3.3, could be used as the arbitrary step-size procedure.
4. Depending on the outcome of these evaluations, specifications should be developed for the selected procedure and the required computer coding accomplished. It should be recognized that it may be desirable to retain the option of using either type of resizing procedure in the final module.
5. Investigation of the flutter optimization task should be extended to include those problems encountered in a realistic design environment which have a direct influence on the performance of an optimization procedure. These problems are discussed in Section 7 and mentioned briefly in the conclusions above. To be of most use, it is felt that such investigations must be made using a structural design task of the complexity of the arrow wing study performed by Lockheed for NASA (NAS-1-12288).
6. The extent to which it is feasible and desirable to integrate the flutter optimization task with the strength optimization task should be investigated with emphasis on means of simplifying the formulation of the strength constraints.

APPENDIX A

NUMERICAL EXAMPLES OF RESIZING PROCEDURES

A.1 INTRODUCTION

To provide the basis for a direct comparison of the several candidate resizing procedures (Section 6) in performing a simplified flutter optimization task, an idealized test case was formulated and numerical evaluations were conducted. Not all candidate procedures were evaluated to the same degree. In most cases the process was discontinued as soon as the results of interest were obtained.

The structural model on which the test case was based is a simple EI, GJ beam representation of a subsonic transport airplane. Although no attempt was made to simulate a realistic design process in detail, the resulting optimization might be regarded as typical of the preliminary design phase of the development of such an airplane. The test case was designed to avoid the difficulties associated with modalization (Section 4) and the nonlinear stiffness effects (Section 7.1) encountered in more practical optimization efforts.

In implementing the various resizing procedures, no specialized computer programs were formed. Instead, existing batch, graphics and remote terminal systems and programs were employed, augmented by hand computations where necessary. As a result, no direct comparison of the computer resources required by the various resizing procedures is available; such information relative to this idealized test case is of little practical significance in any event.

A.2 STRUCTURAL MODEL

The structural model used for the test case is an EI, GJ beam representation of a subsonic transport airplane.

There are 9 grid-points on the wing semi-span, 12 grid-points along the fuselage center line (Figure A-1) and other miscellaneous grid-points used to define a rigid empennage and control surface, for a total of 67 elastic degrees of freedom. Symmetric boundary conditions are imposed. All inertia and aerodynamics coordinates are retained. Eighteen elastic degrees of freedom, not associated with design variables, are eliminated by static stiffness condensation, reducing the number of degrees of freedom to 49.

The design variables are the torsional stiffnesses of the eight structural elements indicated in Figure A-1. The bending stiffness of the wing is not varied. The design variables are defined as increments over values defining a simulated strength design and the associated inertia and stiffness matrices are expressed in terms of a unit mass, so that the total inertia and stiffness matrices are as expressed in equations (A.1) and (A.2).

$$[K] = [K_0] + \sum_{i=1}^8 m_i [\Delta K_i] \quad (A.1)$$

$$[M] = [M_0] + \sum_{i=1}^8 m_i [\Delta M_i] \quad (A.2)$$

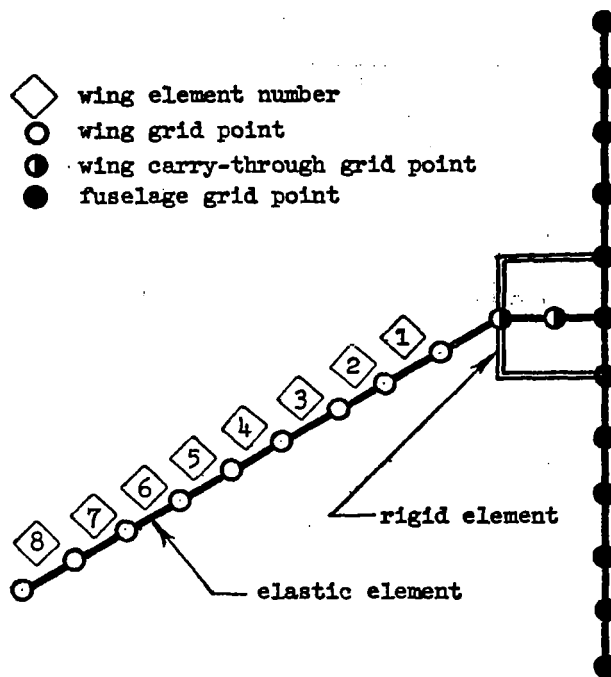


Figure A-1: Structural Model

where $[K_0]$ and $[M_0]$ are the matrices of the fixed stiffness and inertia and $[\Delta K_i]$ and $[\Delta M_i]$ are the stiffness and inertia matrices associated with a unit mass of the design variable m_i . For the present purposes, the minimum-size or strength designed portions of the design variables are included in the K_0 , M_0 matrices, so that the ΔK_i , ΔM_i matrices include only the design variable increments relative to the strength design.

The strength designed configuration is a fictitious configuration defined such that a flutter problem is assured within the design envelope of the airplane. The characteristics of this critical flutter root are shown in Figure A-2, indicating a flutter speed of approximately 226.4 m/s EAS (440 KEAS). Using this as a starting point, the test case required that the flutter speed be increased to 270.1 m/s EAS (525 KEAS) with a minimum of weight increase in the torsional stiffness design variables.

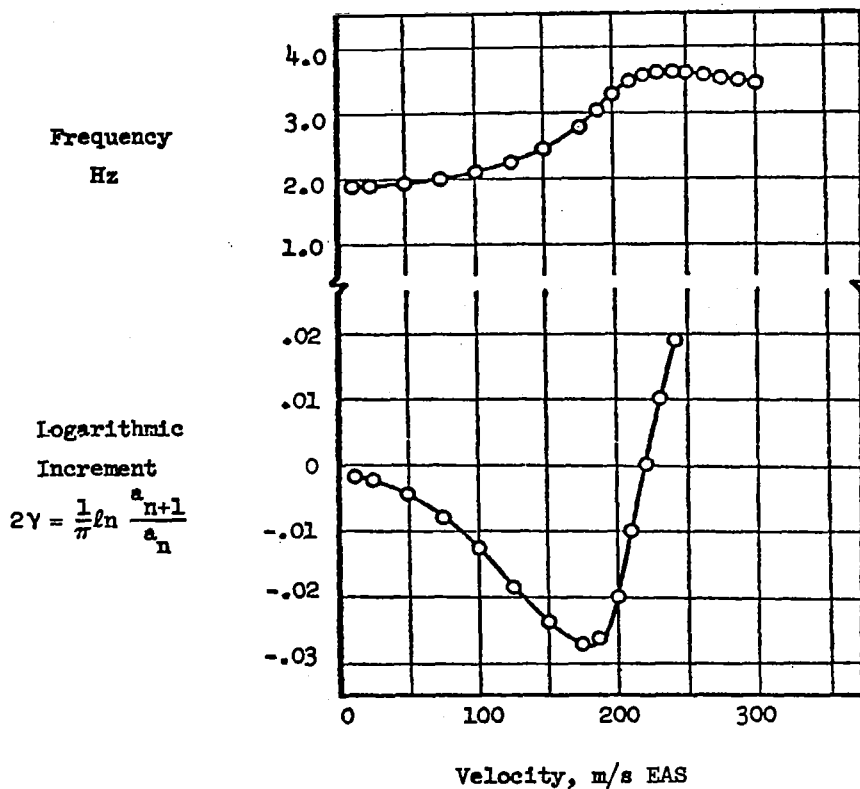


Figure A-2: Frequency and Damping of Critical Flutter Root

In addition to the flutter-deficient (226.4 m/s EAS) configuration, two auxiliary configurations are required. For the constant flutter speed procedures, a configuration having the required flutter speed (270.1 m/s EAS) but a non-optimum distribution of the design variables is needed. This was obtained by increasing the design variables in a manner equivalent to raising the torsional stiffness of the wing by a uniform factor until the required flutter speed was reached. For the penalty function procedure (Reference 16), an initial configuration having a flutter speed in excess of the final flutter speed is required. This configuration was obtained in the same manner as the previous one, except that the flutter speed was increased to 280.4 m/s EAS (545 KEAS). The design variable distributions and total design variable mass for these configurations are shown in Table A-1. A man-in-the-loop resizing procedure, different from any of the methods discussed in Section 6 and using Incremented Flutter Analysis as the principal tool, indicated that the minimum mass for the required flutter speed is approximately 249.6 kg (550.2 lbs). This value is used as a bench mark for further comparisons.

Flutter Speed m/s EAS	Design Variable Mass, kg*								
	1	2	3	4	5	6	7	8	Total
226.4	0	0	0	0	0	0	0	0	0
270.1	113.5	105.2	90.4	65.5	63.6	31.3	21.8	15.3	506.6
280.4	141.3	131.0	112.5	81.6	79.2	38.9	27.1	19.1	630.9

* In accordance with the definition of design variable this is a mass increase above the simulated strength level design.

Table A-1: Initial Configuration; Non-Optimum Distribution

A.3 METHOD OF SIMODYNES

The method of Simodynes is reported in Reference 15 and discussed in Section 6.3 of this report. The numerical evaluations reported here are in fact evaluations, or partial evaluations, of three distinct methods. The Simodynes method itself was evaluated only to the extent of the first resizing cycle. At that point, two undesirable features of the method were identified and a modification of the method was implemented. The numerical evaluation was then continued, using this modified method. In the modified method one of the objectionable features of the original method, the frequency constraint, is avoided, as is discussed in Section 6.3. Finally, a second modification was incorporated by which the influence of the choice of the dependent design variable on the resizing step is eliminated, and an additional numerical evaluation was performed using this procedure.

A.3.1 Original Method of Simodynes - As discussed in Section 6.3, the method of Simodynes is a procedure for minimizing the total weight of a set of design variables while maintaining a fixed flutter speed and frequency. In order to satisfy these constraints, two of the design variable masses are considered to be dependent functions of the remaining design variable masses. The resizing direction is then determined by the gradient of the total weight subject to the flutter speed and frequency constraints.

An initial resizing step was generated for an arbitrarily chosen total mass reduction, W_1 , of 45.4 kg (100 lbs) for each of three pairs of dependent design variables: 1 and 2, 4 and 5, and 7 and 8. The resulting values of the design variable increments are shown in Table A-2.

Dependent Design Variables	Design Variable Increments, kg								Total
	1	2	3	4	5	6	7	8	
1 and 2	-198.6	155.9	-0.0	0.0	-0.0	-0.3	-1.0	-1.4	-45.4
4 and 5	-0.2	-0.2	-0.2	-106.4	68.3	-0.5	-2.3	-3.9	-45.4
7 and 8	-10.1	-10.0	-9.7	-9.0	-7.6	-4.3	20.6	-15.3	-45.4

Table A-2: Initial Design Variable Increments (first resizing step)
for Three Pairs of Dependent Design Variables

The dual constraint of flutter speed and frequency leads to relatively large positive and negative increments in the dependent design variables. When design variable pairs 1, 2 and 4, 5 are chosen as dependent design variables the magnitude of the negative increment is larger than the available amount (cf. Table A-1). The question arises whether to invoke the minimum size constraint and accept the resulting larger drift in flutter speed, or recognize the violation of the minimum size constraint as a temporary situation that will be corrected in subsequent steps, or reduce W_1 .

It should be noted, however, that as a result of the large increments of the dependent design variables, the amount of resizing towards the goal of minimum total mass occurring in the independent variables is small when the pairs 1, 2 and 4, 5 are the dependent design variables. How serious a drawback is implied by these considerations has not been pursued in detail, since it is believed that the frequency constraint by itself puts Simodyn's method at a distinct disadvantage, certainly in view of the rather straightforward manner in which this constraint can be removed, as is demonstrated in the modified Simodyn's method which is the subject of the next section.

A.3.2 Modified Simodyn's Method - The method of Simodyn's was modified to eliminate the flutter frequency constraint, substituting flutter frequency as a dependent variable in place of one of the two dependent design variables used in the original method (Section 6.3).

As in the case of the original method, an initial resizing step was generated for a total mass reduction of $W_1 = 45.4$ kg (100 lbs) for each of three choices of the dependent design variable. The results, shown in Table A-3, indicate more reasonable distributions of the resizing increments due to the elimination of the frequency constraint, although the sensitivity to the choice of dependent design variable remains. Specifically, the result of choosing as a dependent design variable one for which $\frac{\partial V}{\partial m}$ is small in

Dependent Design Variable	Design Variable Increments, kg								Total
	1	2	3	4	5	6	7	8	
1	-50.1	0.0	0.1	0.4	0.9	1.4	1.4	0.4	-45.4
5	-12.8	-12.0	-10.8	-6.1	-14.2	8.5	8.8	-6.8	-45.4
8	-3.4	-3.0	-2.3	0.4	3.9	8.8	9.0	-58.8	-45.4

Table A-3: Initial Design Variable Increment (first resizing step)
for Three Dependent Design Variables; Modified
Simodynes Method

magnitude is demonstrated in Table A-3. With number 1 as the dependent variable, a large negative increment of that variable is required to balance rather small increments in the independent design variables in keeping the

flutter speed constant on a linear basis. For a given weight decrement W_1 ,

the amount of resizing towards the goal of minimum total mass occurring in the independent variables is small. It would seem, therefore, that it is important to choose as a dependent design variable one that corresponds to

a large value $\frac{\partial V}{\partial m}$. With the proper choice of this variable, it would appear

that a reasonably efficient procedure might result. In order to assess this, additional resizing steps were executed, using design variable 5 as the independent variable. The flutter speeds and design variable distributions for six resizing steps are presented in Table A-4. One other minor departure from the original method is that the flutter speed was adjusted to approximately 270.1 m/s EAS at the beginning of each resizing cycle by a uniform percentage increase in the current design variable distribution.

It is believed that the efficient performance indicated by the results shown in Table A-4 may be somewhat better than might typically be achieved by this method. These evaluations were performed relatively late in the contract effort, so that much experience with this test case had been accumulated. This allowed a choice of total mass reduction steps which minimized the required number of steps to achieve the mass reductions shown. Specifically,

the successive values of W_1 chosen (90.7, 90.7, 90.7, 45.4, 9.1, 9.1 kg) anticipate the minimum weight.

A.3.3 Improved Move Vector - To eliminate the sensitivity of the modified method to the choice of dependent design variable, a modified move vector was formulated and tested in a further evaluation.

Step	Design Variable Mass, kg								Total	Flutter Speed m/s EAS
	1	2	3	4	5	6	7	8		
0	113.5	105.2	90.4	65.5	63.6	31.3	21.8	15.3	506.6	270.1
1	88.0	81.1	68.8	53.4	35.2	48.3	39.3	1.7	415.9	267.2
2	60.7	55.1	44.7	38.5	71.4	48.9	35.4	0.0	354.7	268.3
3	24.6	22.2	17.2	30.5	57.2	63.0	48.1	15.9	278.9	267.8
4	6.1	5.5	3.6	26.3	89.8	64.5	47.2	6.1	249.1	268.6
5	0.6	1.0	0.4	27.5	93.8	68.1	49.8	7.6	248.8	269.7
6	0.0	0.0	0.0	28.2	91.3	69.9	51.2	8.6	249.2	270.0

Table A-4: Mass Reductions with Modified Simodynes Procedure

The new move vector is shown in equation (A.3), where $\frac{\partial V}{\partial m_i}$ is the derivative of the flutter speed with respect to the design variable m_i . The first, positive, component of this move vector will be recognized as the

$$\left\{ \Delta m_i \right\} = a \left\{ \frac{\partial V}{\partial m_i} \right\} - b \left\{ \frac{1}{\partial V / \partial m_i} \right\} \quad (A.3)$$

$$a = \frac{W_2}{\left[1 \right] \left\{ \partial V / \partial m_i \right\}} \quad (A.4)$$

$$b = \frac{a \left[\partial V / \partial m_i \right] \left\{ \partial V / \partial m_i \right\}}{\left[\partial V / \partial m_i \right] \left\{ \frac{1}{\partial V / \partial m_i} \right\}} \quad (A.5)$$

velocity gradient. The scalar "a" is determined from equation (A.4), where W_2 is the total mass specified for the positive component of the vector $\left\{ \Delta m_i \right\}$. The scalar "b" defines the magnitude of the negative component of the vector $\left\{ \Delta m_i \right\}$ such that the change in flutter speed, on a linearized basis, is zero.

The flutter velocity derivatives, needed for this move vector, are not computed directly in either the Simodynes or modified Simodynes procedure. Instead, partial derivatives of the dependent variables with respect to independent variables are generated. For the modified Simodynes procedure,

considering the dependent design variable, m_u , and one independent design variable, m_i , keeping all other independent variables constant, the condition $V(m_u, m_i) = \text{constant}$ leads to:

$$\frac{\partial m_u}{\partial m_i} = - \frac{\partial V / \partial m_i}{\partial V / \partial m_u} = - \frac{\overline{\partial V}}{\partial m_i} \quad (\text{A.6})$$

where $\frac{\overline{\partial V}}{\partial m_i}$ is a normalized flutter speed derivative. The negative of the complete set of such derivatives is then equivalent to the flutter velocity derivatives normalized to the flutter velocity derivative of the dependent design variable. Examination of equations (A.3), (A.4) and (A.5) shows that the column of resizing increments, Δm_i , remains invariant with normalization of the flutter velocity derivatives. Accordingly, the normalized derivatives indicated in equation (A.6) are used to evaluate the resizing increments shown in equation (A.3).

Using the move vector described here, with a value of $W_2 = 45.4 \text{ kg}$ (100 lbs), the results of eight resizing steps are shown in Table A-5. In comparing these results with the results in Table A-4 and noting that the "improved" move vector requires more steps to reach the minimum total mass, reference is again made to the fact that the results of Table A-4 were generated after considerable experience had been gained with this idealized test case. Specifically, it was found that the value of W_2 chosen affects the convergence of the procedure. Note that the quantity W_2 in the present procedure has no direct relationship to the quantity W_1 used in the modified Simodynes procedure.

To determine the effect of varying W_2 , steps 5, 6 and 7 were repeated with values of W_2 of 90.7, 90.7 and 136 kg, respectively. The results, presented in Table A-6, demonstrate that improved performance of the method could be expected for a more favorable choice of the value of W_2 .

Step	Design Variable Mass, kg									Flutter Speed m/s EAS
	1	2	3	4	5	6	7	8	Total	
0	113.5	105.2	90.4	65.5	63.6	31.3	21.8	15.3	506.6	270.1
1	58.7	67.5	66.3	58.6	64.8	39.4	30.1	6.9	392.3	269.2
2	25.0	43.2	50.7	55.4	68.0	46.8	36.6	6.8	332.8	269.6
3	0.3	24.6	38.5	53.2	71.4	53.1	41.7	7.9	290.7	269.8
4	0.0	7.5	26.9	50.8	74.0	58.0	45.3	8.5	271.1	269.9
5	0.0	0.0	16.5	48.6	76.2	61.9	47.9	8.8	260.0	270.0
6	0.0	0.0	7.0	46.4	78.1	64.8	49.6	8.9	254.8	270.0
7	0.0	0.0	0.0	44.5	79.6	67.2	50.9	9.1	251.2	270.0
8	0.0	0.0	0.0	42.1	80.4	68.5	51.3	8.6	250.8	270.1

Table A-5: Mass Reductions with Improved Move Vector

Step	W_1 kg	Design Variable Mass, kg									Flutter Speed m/s EAS
		1	2	3	4	5	6	7	8	Total	
5	90.7	0.0	0.0	5.0	45.6	77.7	65.1	49.8	8.7	252.0	269.8
6	90.7	0.0	0.0	0.0	40.9	80.2	69.0	51.5	8.3	250.0	270.0
7	136.1	0.0	0.0	0.0	35.1	82.2	71.9	52.0	8.4	249.6	270.0

Table A-6: Mass Reductions with Increased Values of W_2

A.4 GRADIENT METHODS OF RUDISILL AND BHATIA

In Reference 14, Rudisill and Bhatia present a method of generating flutter velocity derivatives and they suggest several resizing procedures using these derivatives. These procedures are discussed in Section 6.2. Two of these procedures, the velocity gradient search and the gradient projection search, were chosen for numerical evaluation and the results are presented in the following sections.

A.4.1 Velocity Gradient Search - The velocity gradient search is used to increase flutter speed by a series of resizing steps in which the increments in the design variables are proportional to the corresponding elements of the velocity gradient. The resulting distribution is not optimum, but is a good initial distribution for procedures in which the total mass is minimized at constant flutter speed. One obvious application of this procedure is in increasing the flutter speed of a flutter-deficient design to the required flutter speed. Starting with the 226.4 m/s EAS configuration of the test case structural model, the flutter speed was increased to approximately 270.1 m/s EAS in five steps (Table A-7). The design variable increments were formed according to equation (A.7), where the nominal velocity increments, ΔV , were

$$\left\{ \Delta m_i \right\} = \frac{\Delta V}{\left[\partial V / \partial m_i \right] \left\{ \partial V / \partial m_i \right\}} \left\{ \frac{\partial V}{\partial m_i} \right\} \quad (A.7)$$

12.9, 10.3, 10.3, 5.1 and 3.1 m/s EAS. The actual velocity increments do not correspond exactly to the nominal increments because of 1) the nonlinear relationship between the increment in flutter speed and increments in the design variables, and 2) the fact that the velocity derivatives, as defined in Reference 14, are not based on matched atmospheric conditions of Mach number, speed and altitude, whereas the flutter equation was solved directly for matched conditions.

To evaluate the effect of the number of resizing steps used to produce a given velocity increment, the magnitude of the original increment was determined which resulted in the same flutter speed of 270.5 m/s EAS in one resizing step. The required mass, 292.3 kg, does not differ greatly from the multi-step result of 286.7 kg, when compared with the optimum total

design variable mass of 249.6 kg pounds (for $V_f = 270.1$ m/s EAS).

Step	Design Variable Mass, kg								Total	Flutter Speed m/s EAS
	1	2	3	4	5	6	7	8		
1	1.1	1.6	2.4	5.7	10.1	16.6	17.8	8.1	63.4	239.0
2	2.6	3.9	6.0	13.6	23.2	33.2	32.6	15.7	130.9	249.9
3	5.0	7.4	11.2	24.6	40.3	51.3	47.7	23.6	211.0	261.1
4	6.7	9.8	14.7	31.5	50.4	60.8	55.4	27.5	250.8	260.9
5	7.9	11.6	17.2	36.2	57.1	66.8	60.1	29.8	286.7	270.5

Table A-7: Design Variable Mass and Flutter Speed; Velocity Gradient Search

A.4.2 Gradient Projection Search - The gradient projection search is a procedure for reducing the total mass of the design variables while attempting to maintain a constant flutter speed. The column of resizing increments is derived from the velocity gradient and the mass gradient as indicated by equation (A.8)

$$\left\{ \Delta m_i \right\} = \lambda_0 \left\{ \left[\frac{\partial M}{\partial m_i} \right] \right\} + \lambda_1 \left\{ \left[\frac{\partial V}{\partial m_i} \right] \right\} \quad (A.8)$$

$$\lambda_1 = - \frac{\left[\frac{\partial V}{\partial m_i} \right] \left\{ \frac{\partial M}{\partial m_i} \right\}}{\left[\frac{\partial V}{\partial m_i} \right] \left\{ \frac{\partial V}{\partial m_i} \right\}} \quad (A.9)$$

where the scalar λ_1 is determined as in equation (A.9). Comparison with equations (A.3), (A.4) and (A.5) shows that this resizing column is similar to that of Section A.3.3 except the mass gradient is used in place of the reciprocals of the flutter derivatives. For comparison with this previous procedure, the product $\lambda_0 \lambda_1$, the coefficient of the velocity gradient, was chosen to give the same 45.4 kg positive increment as before. The results of the first three resizing steps, starting with the 270.1 m/s EAS configuration, are given in Table A-8. Comparison with Table A-5 indicates that the first three steps of the gradient projection search are less effective than the initial step of the procedure of Section A.3.3. The evaluation of the gradient projection search was terminated at this point.

Step	Design Variable Mass, kg								Total	Flutter Speed m/s EAS
	1	2	3	4	5	6	7	8		
0	113.5	105.2	90.4	65.5	63.6	31.3	21.8	15.3	506.6	270.1
1	105.4	97.4	83.3	61.0	62.3	34.5	25.2	10.3	479.4	270.0
2	98.1	90.6	77.1	57.4	61.9	37.8	28.2	7.2	458.3	270.0
3	91.2	84.1	71.3	54.2	61.8	40.7	30.8	5.4	439.5	270.0

Table A-8: Mass Reductions with Gradient Projection Search

A.5 INTERIOR PENALTY FUNCTION METHOD

The interior penalty function method, described in Reference 16 and discussed in Section 6.4, employs a series of unconstrained minimizations of a modified objective function in order to minimize the objective function of interest (usually total mass). The modified objective function, $\phi(m_i)$, is formed by adding penalty terms, reflecting the constraint equations, to the objective function $W(m_i)$ (equation (A.10)). The minimization of the modified objective

$$\phi(m_i) = W(m_i) + \sum_{j=1}^n \frac{r}{g_j(m_i)} \quad (A.10)$$

function is carried out for repeated reductions of the penalty weighting factor, r , until the minimum of the modified objective function approximates the minimum of the objective function. In the present case, the penalty terms represented minimum size (mass) constraints for each of the eight design variables, in addition to the flutter speed constraint. Starting with the 280.4 m/s EAS (545 KEAS) configuration, the initial penalty weighting factors were chosen to produce a total of the penalty terms approximately equal to the objective function. These penalty weighting factors were reduced in five steps to a value which resulted in penalty terms approximately equal to one percent of the minimum value of the objective function (total mass). The move direction was generated using approximate second derivatives with Newton's method (Section 6.4 and Reference 16). The results of the five resizing steps are given in Table A-9.

Step	Design Variable Mass, kg									Modified Objective Function kg
	1	2	3	4	5	6	7	8	Total	
0	141.3	131.0	112.5	81.6	79.2	38.9	27.1	19.1	630.9	1305.0
1	73.5	75.3	76.3	76.9	88.9	52.9	36.8	23.4	504.1	1178.4
2	31.8	32.8	34.1	47.7	82.1	71.0	49.4	14.2	363.1	492.6
3	14.5	15.6	17.7	37.4	81.4	70.5	48.7	9.4	295.1	329.3
4	2.9	3.5	5.4	33.5	83.8	73.4	52.4	8.5	263.4	274.4
5	2.0	2.1	1.8	30.9	84.1	73.5	52.1	8.5	255.1	258.1

Table A-9: Mass Reductions with Penalty Function Procedure

It is recognized that additional resizing steps in which the penalty weighting factor is further reduced would lead to a lower total mass. However, it has been observed (Section 6.4.2) that each step in the penalty function procedure requires approximately the same number of numerical evaluations as three steps in arbitrary step size procedures, so that the number of steps in this numerical evaluation of the penalty function method is sufficient to provide the comparison with the arbitrary step size procedures (Tables A-4 and A-5).

A.6 METHOD OF FEASIBLE DIRECTIONS

The method of feasible directions (Reference 17 and Section 6.5) generates a series of resizing move vectors, each of which is both feasible (does not violate active constraints) and usable (reduces total mass). Each resizing direction is followed until a new constraint is violated, an active constraint is re-encountered or the total mass is minimized. The resizing direction is found using the Simplex procedure to determine an optimum move direction.

In the present case, the 270.1 m/s EAS configuration was taken as the starting point, and minimum size (mass) constraints were imposed on the eight design variables in addition to the flutter speed constraint. The results of the first eight resizing steps are presented in Table A-10.

Step	Design Variable Mass, kg								Total	Flutter Speed m/s EAS
	1	2	3	4	5	6	7	8		
0	113.5	105.2	90.4	65.5	63.6	31.3	21.8	15.3	506.6	270.1
1	98.6	90.3	75.5	50.7	69.3	46.1	36.7	0.5	467.6	273.0
2	54.9	46.6	31.8	7.0	28.1	89.8	80.4	44.1	382.7	270.1
3	23.6	15.3	0.5	38.3	59.4	110.6	49.0	12.8	309.2	273.0
4	11.2	2.9	0.5	47.7	71.1	98.0	61.4	0.5	293.8	273.7
5	0.5	2.9	0.5	36.9	82.5	87.2	61.4	11.2	283.0	274.0
5A	0.5	2.9	0.5	30.8	76.5	81.2	55.3	5.3	253.0	270.1
6	0.5	0.5	0.5	31.9	79.0	78.7	52.8	7.8	251.6	270.1
6A	0.5	0.5	0.5	31.8	78.9	78.5	52.7	7.6	250.8	270.1

Table A-10: Mass Reductions with Feasible Directions Procedure

It should be noted that steps 5A and 6A are so designated because the flutter speed constraint was not active for those steps, and therefore no flutter speed derivatives were required. The resulting move direction was then one of "steepest descent," determined by the mass gradient. These two steps required somewhat less computing resources than did the remainder of the steps, and it therefore seemed reasonable not to identify them as full steps. It should also be noted that 0.5 kg (one pound) was established as the minimum mass (size) allowed for any design variable, rather than zero as in other procedures. This was later determined to be unnecessary as is discussed in Section 6.5.2. Had this not been the case, the total mass in each of the last several steps would have been slightly less.

A.7 AN OPTIMIZATION METHOD USING INCREMENTED FLUTTER ANALYSIS

Reference 18 describes a resizing procedure developed at Lockheed and used primarily in an interactive mode employing graphics displays. This procedure is discussed in Section 6.6.

Although this procedure is presently in an incomplete state of development, it is of interest in that it employs a unique method of determining resizing step size. This is done by a direct minimization of the objective function (weight) using a direction determined by velocity derivatives and the constraints. This minimization results from, and takes into account, the nonlinear relationship between design variable increments and flutter speed. A new direction is generated after each minimization, and the process is repeated until an acceptable approximation of the minimum total mass is obtained.

The results of 4 resizing steps are presented in Table A-11.

Step	Design Variable Mass, kg								Total	Flutter Speed m/s EAS
	1	2	3	4	5	6	7	8		
0	113.5	105.2	90.4	65.5	63.6	31.3	21.8	15.3	506.6	270.1
1	0.0	0.0	0.0	70.1	71.6	44.0	63.5	19.5	268.6	270.1
2	0.0	0.0	0.0	40.3	84.8	78.2	49.4	0.0	251.8	270.1
3	0.0	0.0	0.0	36.0	82.7	73.2	47.7	10.8	250.3	270.1
4	0.0	0.0	0.0	32.7	84.9	72.7	52.9	6.7	249.8	270.1

Table A-11: Mass Reductions with the Method Using Incremented Flutter Analysis

REFERENCES

1. O'Connell, R. F., Hassig, H. J. and Radovcich, N. A., "Study of Flutter Related Computational Procedures for Minimum Weight Structural Sizing of Advanced Aircraft - Supplemental Data," NASA CR-132722, August 1975.
2. Bisplinghoff, R. L., Ashley, H. and Halfman, R. L., "Aeroelasticity," Addison-Wesley Publishing Company, Inc., 1955.
3. Hassig, H. J., "An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration," Journal of Aircraft, Vol. 8, No. 11, November 1971, pp. 885-889.
4. O'Connell, R. F., "Incremented Flutter Analysis," Journal of Aircraft, Vol. 11, No. 4, April 1974, pp. 236-240.
5. Bhatia, K. G., "An Automated Method for Determining the Flutter Velocity and the Matched Point," Journal of Aircraft, Vol. 11, No. 1, January 1974, pp. 21-27.
6. Phoa, Y. T., "A Computerized Flutter Solution Procedure," Computers and Structures, Vol. 3, pp. 195-204, Pergamon Press, 1973.
7. Desmarais, R. N. and Bennett, R. M., "An Automated Procedure for Computing Flutter Eigenvalues," Journal of Aircraft, Vol. 11, No. 2, February 1974, pp. 75-80.
8. Korn, G. A. and Korn, T. M., "Mathematical Handbook for Scientists and Engineers," Second, enlarged and revised edition, McGraw-Hill Book Company, 1968.
9. Stroud, W. J., Dexter, C. B. and Stein, M., "Automated Preliminary Design of Simplified Wing Structures to Satisfy Strength and Flutter Requirements," NASA TN D-6534, December 1971.
10. Hurty, W. C., "Dynamic Analysis of Structural Systems Using Component Modes," AIAA Journal, Vol. 3, No. 4, April 1965, pp. 678-685.
11. Courtney, R. L., "Finite-Element Modeling Studies in the Normal-Mode Method and Normal-Mode Synthesis," NASA TN D-6326, May 1971.
12. Haftka, R. T. and Yates, E. C. Jr., "On Repetitive Flutter Calculations in Structural Design," AIAA Paper 74-141, AIAA 12th Aerospace Sciences Meeting, Washington, D.C., January 30 - February 1, 1974.
13. Watkins, C. E., Runyan, H. L. and Woolston, D. S., "On the Kernel Function of the Integral Equation Relating the Lift and Downwash Distributions of Oscillating Finite Wings in Subsonic Flow," NACA Report 1234, 1955.

14. Rudisill, C. S. and Bhatia, K. G., "Optimization of Complex Structures to Satisfy Flutter Requirements." AIAA Journal, Vol. 9, No. 3 August 1971, pp. 1487-1491.
15. Simodynes, E. E., "Gradient Optimization of Structural Weight for Specified Flutter Speed," Journal of Aircraft, Vol. 11, No. 3, March 1974, pp. 143-147.
16. Haftka, R. T., "Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements," NASA TN D-7264, July 1973.
17. Gwin, L. B. and McIntosh, S. C., Jr., "A Method of Minimum Weight Synthesis for Flutter Requirements," AFFDL-TR-72-22, Parts I and II, June 1972.
18. Radovcich, N. A., "Structural Optimization with Flutter Speed and Minimum Gage Constraints," Lockheed-California Company Report LR 26405, April 15, 1974.
19. Rudisill, C. S. and Bhatia, K. G., "Second Derivatives of the Flutter Velocity and the Optimization of Aircraft Structures," AIAA Journal, Vol. 10, No. 12, December 1972, pp. 1569-1572.
20. Fiacco, A. V. and McCormick, G. P., "Nonlinear Programming: Sequential Unconstrained Minimization Techniques," John Wiley and Sons, Inc., 1968.
21. Bracken, J. and McCormick, G. P., "Selected Applications of Nonlinear Programming," John Wiley and Sons, Inc., 1968.
22. Fletcher, R. and Powell, M. J. D., "A Rapidly Converging Descent Method for Minimization," Computer Journal, Vol. 6, 1963, pp. 163-168.
23. Haftka, R. T. and Starnes, J. H., Jr., "WIDOWAC (Wing Design Optimization with Aeroelastic Constraints): Program Manual," NASA TM X-3071, October 1974.
24. Zoutendijk, G. G., "Methods of Feasible Directions," Elsevier Press, Amsterdam, Netherlands, 1960.
25. Vanderplaats, G. N. and Moses, F., "Structural Optimization by Methods of Feasible Directions," Computers and Structures, Vol. 3, pp. 739-755, 1973.
26. Hadley, G., "Linear Programming," Addison-Wesley Publishing Company, Inc., 1962.
27. Vanderplaats, G. N., "CONMIN-A FORTRAN Program for Constrained Function Minimization User's Manual," NASA TM X-62,282, August 1973.

28. O'Connell, R. F., Radovcich, N. A. and Hassig, H. J., "Structural Optimization with Flutter Speed Constraints using Maximized Step Size," AIAA Paper 75-778, AIAA/ASME/SAE 16th Structures, Structural Dynamics and Materials Conference, Denver, Colorado, May 27-29, 1975.
29. Wilkinson, K., Lerner, E. and Taylor, R. F., "Practical Design of Minimum Weight Aircraft Structures for Strength and Flutter Requirements," AIAA Paper No. 74-986, AIAA 6th Aircraft Design, Flight Test and Operations Meeting, Los Angeles, California, August 12-14, 1974.
30. Röhrle, M., "Frequency-Dependent Condensation for Structural Dynamic Problems," Journal of Sound and Vibration, Vol. 20, No. 3, 1972, pp. 413-414.
31. Guyan, R. J., "Reduction of Stiffness and Mass Matrices," AIAA Journal, Vol. 5, No. 1, February 1965, p. 380.
32. Benscoter, S. U. and MacNeal, R. H., "Equivalent Plate Theory for a Straight Multicell Wing," NACA Technical Note 2786, September 1952.
33. Stroud, W. J., "Automated Structural Design with Aeroelastic Constraints: A Review and Assessment of the State Of The Art," in AMD-Vol. 7: Structural Optimization Symposium; pp. 77-118. The Winter Annual Meeting of the American Society of Mechanical Engineers, New York, N.Y., 17-21 November 1974.
34. McCullers, L. A. and Lynch, R. W., "Composite Wing Design for Aeroelastic Requirements," Proceedings of the Conference on Fibrous Composites in Flight Vehicle Design, Dayton, Ohio, 26-28 September 1972, AFFDL-TR-72-130, pp. 951-972.
35. Razani, R., "The Behavior of the Fully-Stressed Design of Structures and its Relationship to Minimum-Weight Design," AIAA Journal, Vol. 3, No. 12, December 1965, pp. 2262-2268.
36. Whetstone, W. D., "The SPAR System," presented at the Integrated Structural Analysis System Symposium, 2 May 1974, NASA, Langley Research Center, Hampton, Virginia.