Application of Optimization Technology to Wing/Store Flutter Prediction

Richard R. Chipman* and John B. Malone*
Grumman Aerospace Corporation, Bethpage, N. Y.

Gradient-directed numerical search techniques, using derivatives of flutter speed with respect to store parameters, are applied to the problem of determining critical flutter configurations for a wing with multiple external stores. The elementary steepest-descent method is first applied to demonstrate the feasibility of the overall approach; a more efficient and reliable algorithm (rank-one-correction) is later introduced to obtain a satisfactory search procedure. Applications to a two- and a four-variable problem are presented. The method offers a useful alternative to current practices and can reduce the potentially catastrophic possibility of "missing" critical store configurations.

Nomenclature

C = [see Eqs. (12) and (14)]
$\{d\}$ = direction of move
f = normalized flutter speed [see Eq. (16)]
$\{g\}$ = gradient vector ($g_i = \partial V/\partial x_i$)
$\{G\}$ = normalized $\{g\}$ [see Eqs. (5) and (17)]
$[H]$ = inverse of Hessian matrix (the matrix of second partial derivatives of V)
M_1, I_1 = mass and pitch moment of inertia of outboard store
M_2, I_2 = mass and pitch moment of inertia of inboard store
$P(\{d\})$ = projection of vector $\{d\}$ onto the active constraints
s_i = scale factor [see Eq. (4)]
V = flutter speed
ΔV_R = requested change in V per step
$\{x\}$ = vector of a combination of store parameters
$\{X\}$ = normalized x [see Eqs. (4) and (15)]
$\{\Delta X\}$ = change or move in X
$\{Y\}$ = change in G [see Eq. (11)]
$\{Z\}$ = [see Eq. (10)]
α = step size
δ_i = range in variable x_i
1* = first minimum
2* = second minimum

Introduction

THE increasingly large inventory of wing-mounted external stores carried by current attack aircraft necessitates the development of safer and more efficient methods for determining wing-store flutter prevention design criteria. The methods must be safer in the sense of minimizing the risk of "missing" flutter-critical configurations from the millions of combinations of stores that must be considered, and they must be more efficient in terms of reducing computer usage and the effective calendar time required to formulate the design requirements.

The most recent developments in wing-store flutter prevention have been directed mainly toward increasing the efficiency of flutter speed computations for variations in the design parameters – the so-called fast-flutter techniques of Ferman[1] and Cross and Albano.[2] Since both these techniques significantly speed up the process of establishing flutter-speed/store parameter trends, they ultimately result in design requirements that are determined from a broader analysis data base than has been feasible in the past. However, the selection of analysis baseline data, from which critical cases are determined, remains a somewhat arbitrary process in which judgement of the flutter analyst is an all-important factor. Furthermore, later, when the aircraft is in the fleet, the addition of a new store to the inventory may necessitate a modification of or addition to the set of critical cases and the performance of accompanying additional flutter analyses. Again, this selection of critical configurations is essential to flutter prevention and is presently being based on engineering judgement.

Although judgment will continue to play an essential role in any approach to wing-store flutter prevention, it is apparent that the risk of missing critical design configurations can be reduced by a more mechanized approach to this complex multiparametric search problem. This paper presents such an approach, employing numerical search procedures common to optimization technology.

Also called nonlinear programming methods or mathematical programming techniques, numerical search procedures have been used in structural design for over a decade – chiefly for the purpose of determining near-minimum-weight structures satisfying strength requirements. The work of Gellatly et al.[3] is an early example of such work. More recently these methods have been applied to designing structures with aeroelastic constraints as well. Gwin and Taylor[4] and Haftka[5] give successful illustrations of this application, and Stroud[6] gives an extensive review of the work in this area. The success of numerical search procedures in all of these usages motivated the present application to the store-flutter problem.

Theory

The basic approach is to regard the determination of critical store-flutter configurations as an optimization problem in which the combinations of permissible store parameters which result in minimum flutter speeds are to be found. Because the discrete combinations of store parameters constituting the inventory that may be carried by an aircraft are quite numerous, the space of parameters is assumed to be a continuum and is searched by standard numerical techniques to determine points of minimum flutter speed. We will consider the problem to be characterized by a fixed set of store carriage locations at each of which the store parameters may vary. Typical parameters might be store mass, moments of inertia and center-of-gravity location, and pylon flexibilities. The limits defining the ranges of these variables

Received Feb. 7, 1977; presented as Paper 77-453 at the AIAA Dynamics Specialists Conference, San Diego, Calif., March 24-25, 1977; revision received Jan. 30, 1978.

Index categories: Aeroelasticity and Hydroelasticity; Structural Dynamics.

*Senior Engineer, Aeroelasticity Methods. Member AIAA.

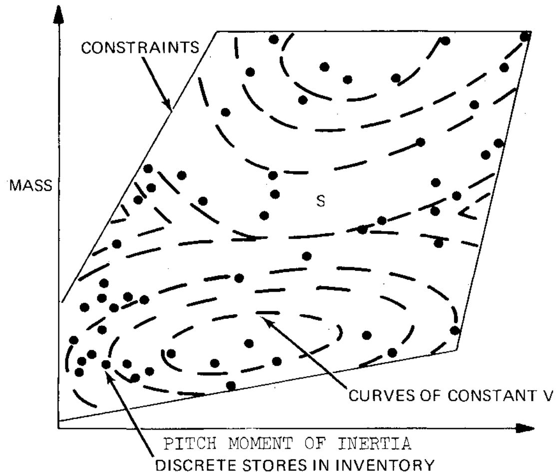

Fig. 1 Basic approach for determining critical stores.

are the simple constraints on the problem. Figure 1 illustrates the main points of this conceptualization. In the present study, attention is confined to store mass and pitch moment of inertia parameters; and, unlike the sketch in Fig. 1, the constraints are assumed to be constants representing lower and upper bounds on the variables.

Search Algorithms

On the assumptions that the design space possesses only a limited number of discontinuities and is continuous and smooth throughout most of its extent, gradient-directed search algorithms are employed in this study. Starting from an arbitrary point, $\{X^0\}$, which represents a feasible combination of store parameters, each search algorithm generates a series of steps to decrease the flutter speed, V. The process is continued until the search is unable to generate another step which appreciably lowers V; if this condition is accompanied by near-zero derivatives of V with respect to all variables which have not reached their constrained values, the search is said to have converged to a minimum. Each search algorithm is of the form

$$\{\Delta X^n\} = \alpha_n \{d^n\} \tag{1}$$

$$\{X^{n+1}\} = \{X^n\} + \{\Delta X^n\} \qquad (n=0,1,\ldots) \tag{2}$$

Steepest Descent

To demonstrate the feasibility of the basic approach, the elementary steepest descent algorithm was first implemented. In this method, $\{d\}$ is chosen to be the direction which decreases the function most locally, namely, the negative gradient

$$\{d^n\} = -\{G^n\} = -\{\nabla V^n\} \tag{3}$$

We scale each variable by its total allowable range divided by the range of the variable with the smallest total range

$$X_i = x_i/s_i = x_i\delta_j/\delta_i \tag{4}$$

Consequently,

$$G_i = s_i g_i \tag{5}$$

The step size, α_n, is chosen to attemp to obtain, in the linear regions of space, a user-requested decrease in flutter speed, ΔV_R. Hence,

$$\Delta V_R = \{\Delta X^n\}\cdot\{G^n\} = -\alpha_n\{G^n\}\cdot\{G^n\} \tag{6}$$

$$\alpha_n = -\Delta V_R/(\bar{G}^n)^2 \tag{7}$$

To avoid overly large steps, each move is truncated, if necessary, so that the change in each variable does not exceed a given percentage of its range.

At boundaries, gradient projection[7] is used to constrain the move to the feasible space. Specifically a move is made along the direction of the projection of the negative gradient, $P(-\{G^n\})$. To retain the desired change in velocity, ΔV_R, the step size is increased by neglecting in Eq. (7) the component of $\{G^n\}$ normal to the constraint.

Rank-One-Correction Scheme (ROC)

To obtain a reliable search procedure, it is well known that one must abandon the steepest descent approach in favor of one of the many algorithms based on conjugate directions, such as the Davidon algorithm.[8] Recent work in numerical methods has developed a hybrid of these methods that is superior in that there is no requirement for accurate line searches to choose a proper step size. In the present application, function evaluations (flutter-speed solutions) are extremely expensive, requiring multiple solutions of complex eigenvalue problems; but the additional cost of gradient evaluations is minimal. Consequently, the absence of this requirement should make this hybrid well suited to our problem. Thus, we have chosen one such algorithm (rank-one-correction[9]) for the present study.

The direction of the search is given by

$$\{d^n\} = -[H_n]\{G^n\} \tag{8}$$

where $[H_n]$ is an estimate of the inverse of the local Hessian matrix and is generated by the series

$$[H_n] = [H_{n-1}] + (1/C)\{Z^n\}\{Z^n\}' \tag{9}$$

$$\{Z^n\} = \{\Delta X^{n-1}\} - [H_{n-1}]\{Y^n\} \tag{10}$$

$$\{Y_n\} = \{G^n\} - \{G^{n-1}\} \tag{11}$$

$$C = \{Z^n\}\cdot\{Y^n\} \qquad (\text{for } n=1,2,\ldots) \tag{12}$$

The series is initiated by

$$[H_0] = [I] \tag{13}$$

If, in a particular step, the use of Eq. (12) would lead to the destruction of the positive definiteness of $[H]$, the following choice of C is made:

$$C = \{Z^n\}\cdot\{Z^n\} \tag{14}$$

Each variable is scaled by its total allowable range and the function, flutter speed, by its maximum expected value

$$X_i = x_i/\delta_i \tag{15}$$

$$f = V/V_{\max} \tag{16}$$

Consequently,

$$G_i = g_i\delta_i/V_{\max} \tag{17}$$

A step size of $\alpha_n = 1.0$ is used on all steps subject to the condition that the resulting step is not unduly large; i.e.,

$$|\{X^n\}| < 0.3 \tag{18}$$

If necessary, the step size is repeatedly halved until this condition is satisfied.

To adapt the algorithm to constrained space, we have used a procedure analogous to gradient projection. If a move computed using the direction $\{d^n\}$ of Eq. (8) would lead into the infeasible region, it is projected back onto the constraints.

There is no guarantee, however, that this projection $P(\{d^n\})$ will be in a direction which allows a function decrease. Consequently, the sign of $\{G^n\}\cdot P(\{d^n\})$ is examined; and, if it is positive, a move equal to the projection of the negative gradient, $P(-\{G^n\})$, is made instead of $P(\{d^n\})$. In either case, the Hessian is updated by Eq. (9).

Special Features

Minimal Line Search

Most gradient-directed algorithms employ a line search to determine an optimal step size, α_n, which results in the maximum decrease in the objective function obtainable by moving from $\{X^n\}$ in the direction $\{d^n\}$. Because the performance of this search requires many additional function evaluations, which are very time consuming in our case (flutter solutions), only a minimal line search is employed. If a move made with a given step size results in a point with a lowered V, the point is accepted and no line search is made; if V is not lowered, the search is conducted. The search consists of performing a polynominal fit in V using values of the function and its gradient at the stepoff point, $\{X^n\}$, and the rejected, overshot point, $\{X^{n+1}\}$. The point at which this polynomial is minimum is then taken as $\{X^{n+2}\}$. Specifically, for the rank-one-correction scheme, a cubic fit is performed using V^n, $\{G^n\}$, V^{n+1}, $\{G^{n+1}\}$.

Direct Search

Due to changes in flutter mechanisms, discontinuities or near-discontinuities can occur in flutter speed. Obviously, gradient-directed searches can fail in such regions. Consequently, we have incorporated logic to detect possible discontinuities and to take remedial action in the form of a direct search. A design point is considered to be near a discontinuity or a local minimum whenever the line search has failed twice in succession (i.e., when α has been reduced twice and V still has not been lowered). When this occurs, the magnitude of each component of the gradient vector is examined. If all components are small, proximity to a local minimum is assumed and additional line searches are made until a successful step is obtained. If one or more of the gradient components is larger than a specified tolerance, the presence of a discontinuity is assumed. In this case, a direct search is attempted to locate a nearby point in space with a lower flutter speed. The direct search is conducted by varying one parameter per cycle by a user-specified percentage.

Termination Criteria

In the present study, a search is considered to be converged when the following three conditions are met:

$$V^n - V^{n-1} \leq 0.5\text{knot} \tag{19a}$$

$$|\{X^n\}-\{X^{n-1}\}| \leq 0.01 \tag{19b}$$

$$|\{G^n\}| \leq 0.0125 \tag{19c}$$

(excluding components normal to active constraints). An exception to these criteria is made for two study search cases (D and E described below). For the purpose of creating demanding test cases, the criteria are tightened for these runs.

$$V^n - V^{n-1} \leq 0.1\text{knot} \tag{20a}$$

$$|\{X^n\}-\{X^{n-1}\}| \leq 0.002 \tag{20b}$$

$$|\{G^n\}| \leq 0.003 \tag{20c}$$

Computer Implementation

A pilot computer program was developed which has been given the acronym ESP (external-stores search procedure). ESP is built around an extensive analysis and structural optimization package, flutter and strength optimization program (FASTOP[10]). In particular, the flutter optimization module (FOP) of FASTOP is used to perform three tasks: 1) vibration analysis, 2) flutter analysis, and 3) computation of the derivatives of flutter speed with respect to store parameters. Modifications to FOP were made to enable it to perform task 3 and to allow the multiple, sequential performance of all three tasks necessary for implementing the iterative search. Additional routines were written to direct the search and change the store parameters during the search process.

An expression for the derivatives of flutter velocity, V, with respect to design variables, has been derived by Rudisill and Bhatia,[11] and involves derivatives of the inertia and stiffness matrices for the problem. For the current study, the design variables are just the inertias of each store; and the derivatives of the stiffness matrix with respect to these variables at zero. Expressions for the derivatives of the inertia matrix are nonzero and are derived in Ref. 12.

Study Problem

To evaluate the methods, a simple study problem was constructed and analyzed at a constant Mach number of 0.8, using the doublet-lattice method of Ref. 13 to compute unsteady aerodynamics of the wing and a modified version of the *p-k* method of Ref. 14 to perform the flutter solutions. The aerodynamics of the stores were neglected.

Description of Study Model

The study model is a two-cell all-aluminum wing box, modeled with finite elements. Membrane elements represent the wing covers; shear panels represent the spar and rib webs; and bar elements are introduced between upper and lower cover node points. The wing root is built in by fully constraining all structural nodes on the root boundary. Two store stations are included and the pylon at each is modeled by beam elements.

The dynamic idealization consisted of 27 nodes and 49 degrees of freedom. Each store was allowed three translational and two rotational (pitch and yaw) degrees of freedom. During all analyses, the pitch and yaw moments of inertia were set equal to each other. Mode shapes for the wing with representative stores attached were characterized by large relative motion of the store degrees of freedom.

Description of Study Space

Two spaces of store parameter variations were investigated. In the first, the parameters of the outboard store station were held fixed at $M_1 = 500$ lb and $I_1 = 200{,}000$ lb-in.^{2} while the mass and pitch inertia of the inboard store were allowed to vary over the ranges ($200 \leq M_2 \leq 2000$ lb) and ($25{,}000 \leq I_2 \leq 500{,}000$ lb-in.^{2}). This two-variable problem facilitated visualization of the space and the progress of searches initiated in it. Figure 2 represents a mapping of contours of constant flutter speed in this space. Several minima Ⓛ, maxima Ⓗ, saddle points Ⓢ, and a line of near-discontinuity (at $M \approx 850$ lb) are seen to exist.[12] The second study space was formed by allowing the mass and pitch inertia of both stores to vary—a four-variable problem—over the ranges ($200 \leq M_1 \leq 1000$ lb), ($200 \leq M_2 \leq 2000$ lb), and ($25{,}000 \leq I_1, I_2 \leq 500{,}000$ lb-in.^{2}).

Results of Simple Searches

Two-Variable Space

On Fig. 2 the progress of several steepest-descent searches initiated from various points in the two-variable space of store parameters is plotted. In Table 1, the flutter speeds computed at every point of each of these searches are listed. The first three searches (A, B, C) converge to minima along the con-

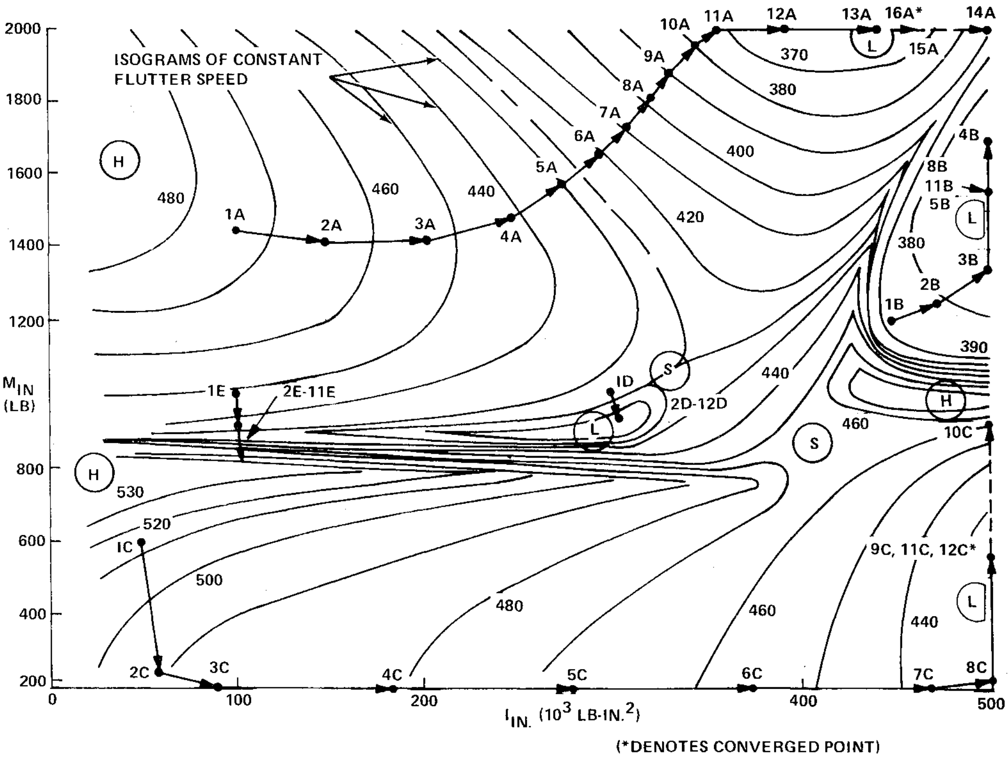

Fig. 2 Two-space mapping and searches using steepest-descent method (M_{out} = 500 lb, I_{out} = 200,000 lb-in.^{2}).

straint boundaries (non-interior minima). Using $\Delta V_R = -10$ knots in the computation of step size, the first search (A) progresses smoothly and converges to a minimum at point 16. Notice that both the direction and size of the steps vary as the search progresses and that the projection of the desired steps are made at the boundary (points 11-16). Starting from another point, search B converges to a different minimum in 11 cycles. Using $\Delta V_R = -20$ knots (twice that of runs A and B) search C traverses a large distance along two different boundaries in reaching a third minimum.

Searches D and E were intended primarily to test the ability of the algorithms to converge to an interior minimum. However, because the only such minimum in Fig. 2 occurs in the vicinity of the near-discontinuity, it became necessary to test simultaneously the ability of the algorithms to progress satisfactorily through this difficult region. To avoid termination of these searches prior to locating the minimum accurately in pitch inertia as well as mass, the more stringent set of termination criteria discussed earlier [Eqs. (20)] were used. As indicated in the enlargement given in Fig. 3, search D is characterized by numerous overshoots and subsequent, extensive reliance on the line search, and is unable to negotiate the narrow valley in this region of space. The enlargement shown in Fig. 4 shows that search E also encounters many overshoots. For this run, the direct-search feature was not included, and the search was unable to travel along the discontinuity.

Figure 5 shows the results of search E rerun with the direct-search feature operational. This time, the search is able to progress along the discontinuity. Column one of Table 2 gives the flutter speeds at each point of the search. The search was terminated arbitrarily after eight cycles.

These searches demonstrate the feasibility of the basic approach and highlight the expected inadequacy of steepest descent. The most troublesome searches, D and E, were repeated using the ROC algorithm. Columns two and three of Table 2 list V at each point. Figure 6 shows the path taken by search D. Notice that only two overshoots occur (points 3 and 5) and that the directions taken at points 7 and 8 differ greatly from the negative gradients (indicated by double-tipped arrows). The minimum is now reached in 11 cycles. As shown in Fig. 7, run E using ROC now moves along the near-discontinuity without resorting to the direct search and converges to the actual minimum. It should be noted that special convergence criteria [Eqs. (20)] employed in these runs are very demanding both in the accuracy of the flutter solution and the search procedure.

Four-Variable Space

Searches were initiated in the four-store variable space from nine separate points, using both of the algorithms. Table 3 lists the locations of these starting points and the flutter speed at each. The results of the searches are indicated by the number of cycles run (followed by a superscript (b) for a case where convergence occurred) and the final flutter speed. Using steepest descent, convergence is obtained for three cases. The ROC scheme, however, converges in every case. In all instances the final flutter speeds obtained by ROC are lower than those reached by the other algorithm. With the exceptions of cases 3 and 4, the number of cycles required by ROC are fewer than required by the other algorithm. For

Table 1 Flutter speeds during the steepest-descent searches in sample two-parameter problem

Cycle	Case A	Case B	Case C	Case D	Case E
1	472.4	387.0	510.8	426.2	446.3
2	463.6	376.1	490.9	422.6	430.6
3	452.4	368.4	487.7	438.8	474.0
4	440.8	368.3	480.1	422.1	429.8
5	430.4	365.1	469.8	522.7	550.0
6	420.1	390.8	456.2	422.0	429.5
7	410.0	365.8	435.7	514.7	532.0
8	400.0	364.9	428.1	421.8	429.2
9	390.0	390.8	427.3	505.4	492.0
10	381.7	365.0	459.5	421.7	434.8
11	374.5	364.9	427.4	494.4	429.2
12	365.5		427.3	421.6	
13	359.5				
14	390.8				
15	359.6				
16	359.5				

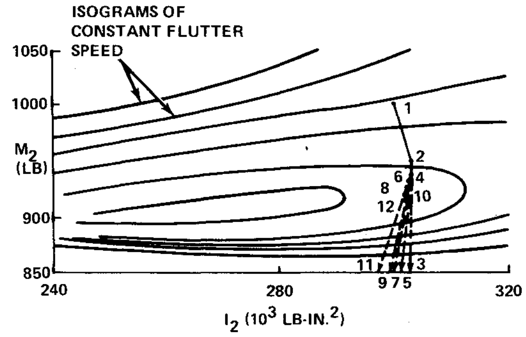

Fig. 3 Search D using steepest-descent method.

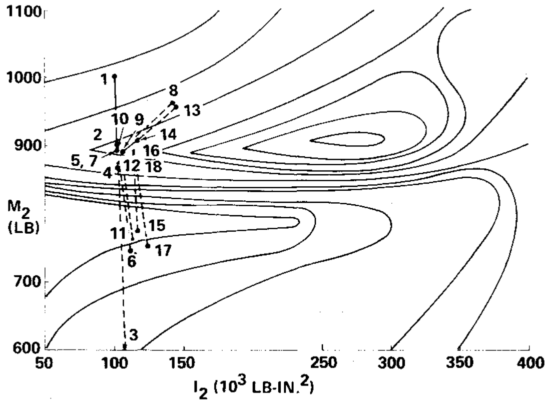

Fig. 4 Search E using steepest-descent method.

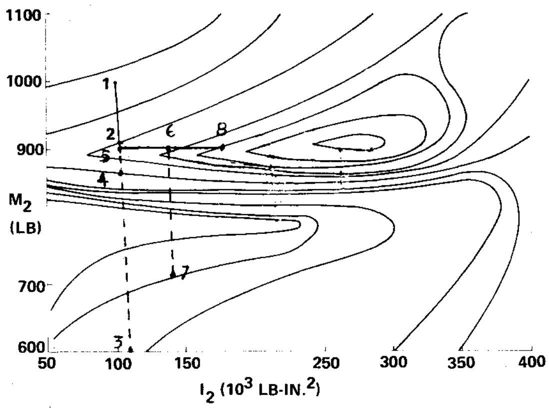

Fig. 5 Search E′ using steepest-descent method with direct search algorithm.

these cases, the higher number of cycles required by ROC is misleading because the algorithm actually terminates at a more critical point than does the steepest descent.

In summary, the results demonstrate the ability of the ROC algorithm to consistently search out minimum flutter speeds; also, as expected, it is both more reliable and more efficient in locating critical combinations of store parameters than steepest-descent.

Convergence Properties of ROC

In Fig. 8, the convergence in flutter speed of the searches made with ROC is inspected in more detail. For each search, the ratio of the incremental flutter speed at each cycle to the final speed for that search is plotted. After 10 and 12 cycles, all speeds are seen to be within 4% and 1% of their final values, respectively. Furthermore, after 12 cycles, the final points are reasonably close to the locations arrived at after the criteria of Eqs. (19) are met. Consequently, somewhat less demanding convergence criteria could have been adopted in this case. This would result in 12 or fewer cycles being required per start to detect a local minimum for this four-design-variable problem.

Table 2 Flutter speeds during one steepest-descent and two ROC searches in sample two-parameter problem

Cycle	Steepest descent Case E′	ROC Case D	ROC Case E
1	446.3	426.2	446.3
2	430.6	422.4	486.0
3	504.4	494.0	429.2
4	529.9	422.3	438.0
5	514.5	423.0	429.1
6	427.5	421.5	429.0
7	514.9	421.4	428.7
8	426.7	421.3	428.3
9		420.80	427.6
10		420.77	426.7
11		420.76	425.5
12			424.3
13			422.9
14			422.5
15			421.3
16			420.86
17			420.9
18			420.76

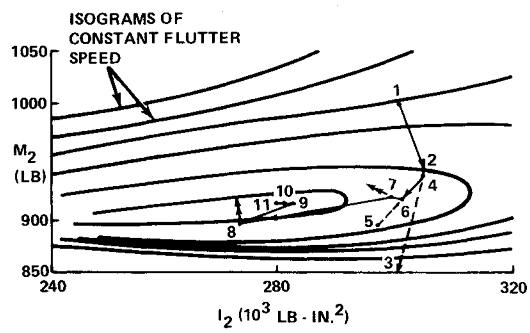

Fig. 6 Search D using ROC.

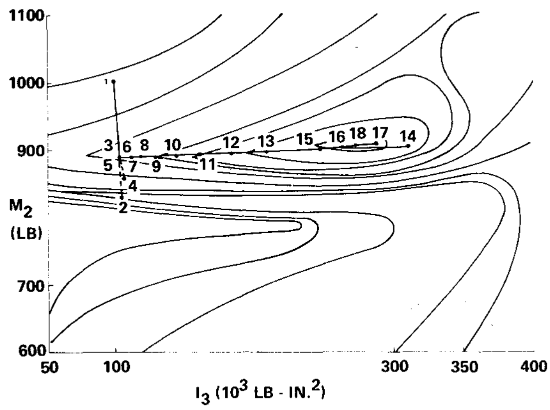

Fig. 7 Search E using ROC.

Critical Configurations as Determined by ROC

For the ROC searches, Figs. 9a and b show the starting and final values of the outboard- and inboard-store parameters, respectively. Two minima clearly emerge in Fig. 9a: $(M_I, I_I)_{1^*} = (790$ lb, $500{,}000$ lb-in.$^2)$ and $(M_I, I_I)_{2^*} = (625$ lb, $400{,}000$ lb-in$^2)$. To clarify which minimum is associated with run 1, that search was continued beyond the convergence tolerances for several additional cycles. The half-open circle denotes the point reached; the second entry for that run in Table 3 indicates the flutter speed obtained. As seen, run 1 is

Table 3 Summary of searches in sample four-parameter problem

	Starting values					Steepest descent		ROC	
Run	M_I	I_I	M_2	I_2	Initial V, knot	NC[a]	Final V, knot	NC[a]	Final V, knot
1	1000	500,000	2000	500,000	322	19[b]	291.8	11[b]	291.8
								24[c]	283.6
2	750	375,000	1500	375,000	334	23	288.6	13[b]	283.4
3	750	375,000	500	375,000	348	6[b]	286.7	16[b]	276.2
4	750	375,000	500	125,000	355	8[b]	284.1	12[b]	279.0
5	250	375,000	500	375,000	776	24	295.6	10[b]	277.8
6	750	125,000	500	125,000	838	24	293.3	14[b]	278.4
7	250	125,000	1500	375,000	786	19	355.7	15[b]	285.8
8	250	125,000	500	375,000	385	23	309.5	13[b]	281.9
9	250	125,000	500	125,000	420	24	315.8	19[b]	283.4

[a] NC is the number of cycles.
[b] Converged.
[c] Run 1 continued beyond convergence.

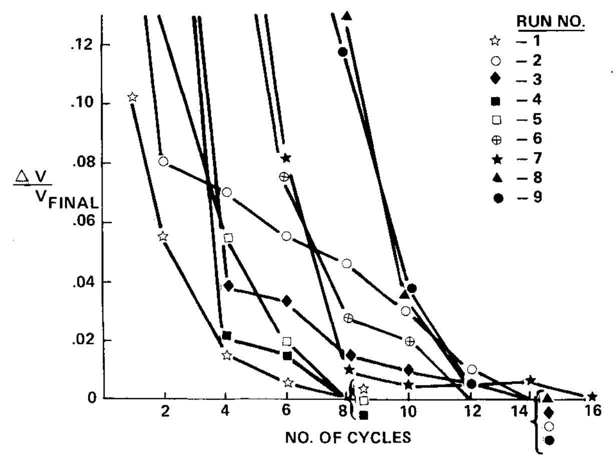

Fig. 8 Convergence of ROC searches in four-parameter problem.

associated with the second minimum. Run 7 could have been continued, if desired, to bring its final point even closer to the second minimum.

Inspection of Fig. 9b reveals that the runs (1, 2, 7, 9) associated with the second outboard-store minimum, $(M_I, I_I)_{2^*}$, converge to roughly the same inboard-store-parameter values, $(M_2, I_2)_{2^*} = (200$ lb, 25,000 lb-in.$^2)$. The runs (3, 4, 5, 6, 8) associated with the first outboard-store minimum, $(M_I, I_I)_{1^*}$, terminate in the inboard-store region $M_2 < 1400$ lb but have not homed-in very precisely. Because Table 3 indicates that these final points are within 6 knots of one another and, as the points have met the convergence criteria of Eqs. (19), this region must be flat; i.e., when the outboard store is of a critical mass and pitch inertia, flutter speed is relatively insensitive to large changes in the inboard store properties. A rough two-space mapping, made with $(M_I, I_I) = (M_I, I_I)_{1^*}$, is presented in Fig. 10 together with all final points, and confirms the above conclusion regarding the flatness of the space below M_2 of 1400 lb and shows the two minima $(M_2, I_2)_{1^*} = (1050$ lb, 2500 lb-in.^{2} and $(M_2, I_2)_{2^*} = (2000$ lb, 2500 lb-in.$^2)$.

In practice, the results typified by this example might be used in the following way: If the minimum flutter speed found by the searches exceeds the speed for which the airplane must be cleared, the required clearance has been demonstrated. In such a case, the most critical configuration as found by the searches might then be wind tunnel or flight tested. If the airplane must be cleared for a higher speed than the minimum detected, a flutter-speed deficiency exists. Three choices are then available. Either the structure must be redesigned, certain store combinations must be excluded from the in-

a)

b)

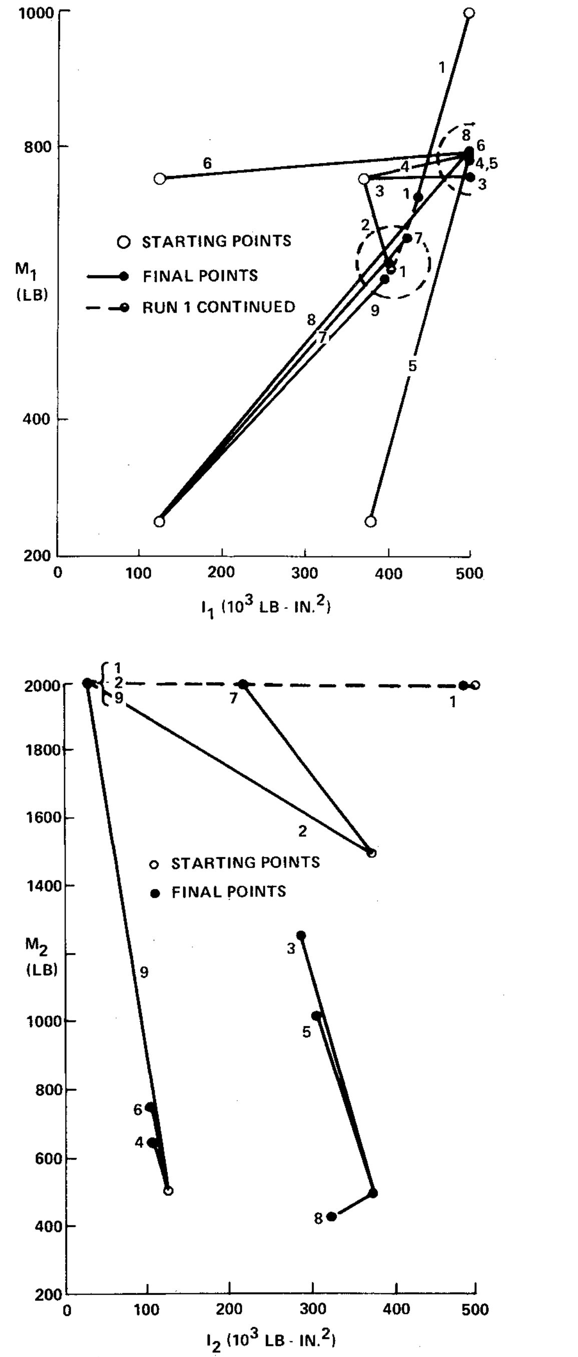

Fig. 9 Progress of ROC searches in four-space mapping: a) outboard-store parameters; b) inboard-store parameters.

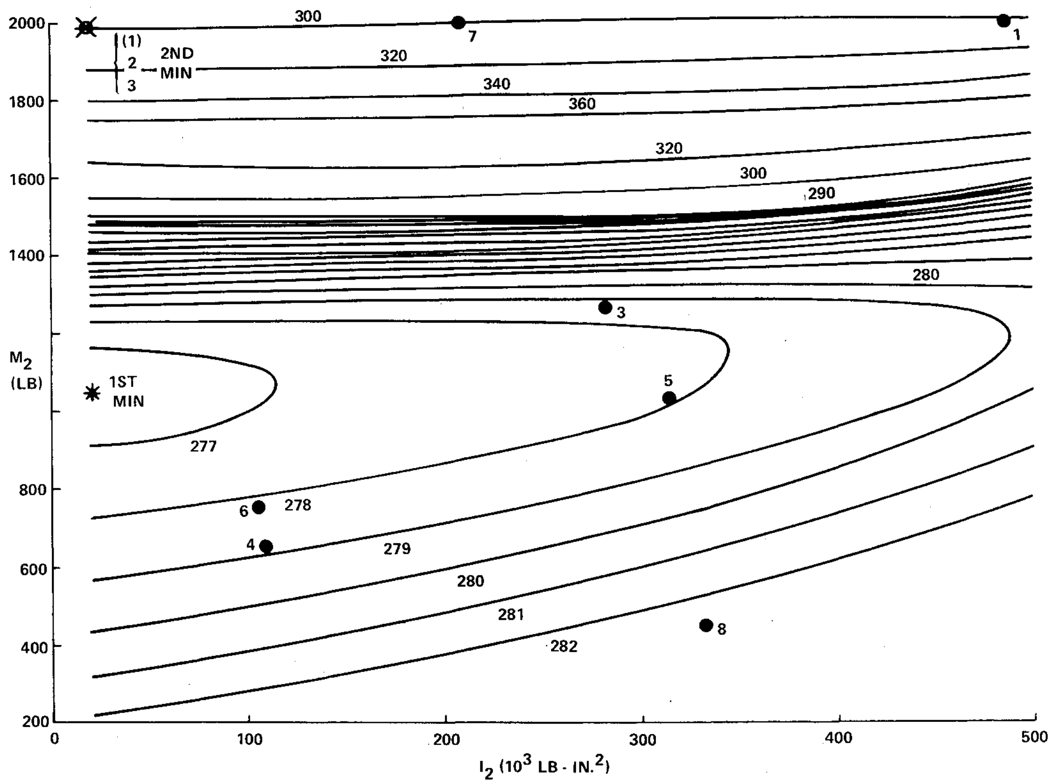

Fig. 10 Mapping of inboard-store parameters with outboard parameters fixed at critical point (M_{out} = 790 lb, I_{out} = 500,000 lb-in.^{2}).

ventory, or speed placards must be placed on the aircraft when carrying certain critical stores. If the first option is chosen, a redesign tool such as FASTOP[10] should be used. If the second option is to be explored, the method provides an approach: As the search converges on the true minima, trend information of near-critical configurations is generated. This information can be used to help construct a reduced inventory by deleting stores in a neighborhood of these critical configurations. This reduced inventory can be searched for new critical configurations and their flutter speeds. If the third option is to be employed, the analyst has no recourse but to generate trend data; however, the searches have localized the areas in which this trend information is needed and have already generated some of this data.

Concluding Remarks

The results of this study lead to the following observations regarding the general application of numerical search techniques to store flutter in contrast to the use of more conventional approaches.

1) The complexity of the space characteristic of the store-flutter problem casts doubt on the validity of simple trend studies now often employed. A comparison of Figs. 2 and 10 demonstrates that trends with certain parameters held fixed at one set of values can be markedly different from those with the same parameters held fixed at another set of values. By using search techniques, however, the locations of critical store combinations are ascertained irrespective of this complexity.

2) The generation of trend information is an expensive proposition. Search techniques obtain the essential information (critical configurations) with many fewer analyses and also provide some trend information as each search progresses. Also the final few cycles of a search are in the vicinity of a critical configuration, where the trend data obtained is of most interest.

3) Because of the high cost of conventionally generating trend information, the cases normally explored are far fewer than necessary to identify critical configurations confidently. Although numerical searches may also occasionally miss a minimum (if an insufficient number of starting points are picked), the confidence level using this approach is much higher.

A comparison of the search algorithms investigated leads to the following conclusions.

1) Gradient-directed searches can be used effectively to determine flutter-critical combinations of store parameters.

2) The rank-one-correction algorithm is reliable and efficient for this application.

3) For a four-design-variable problem, 12 or fewer cycles of the ROC search were adequate. In the cases studied, convergence to within 1% in flutter speed was obtained within 12 cycles.

This investigation has demonstrated the effectiveness and efficiency of the numerical search approach in determining critical store-flutter configurations. To develop ESP to its full potential, the following improvements and extensions are required.

1) ESP should be expanded to include pylon flexibilities and store center-of-gravity location as search parameters. This will enable complicated store racks (TER's and MER's) to be handled.

2) The representation of store inventory limits as constant constraints should be upgraded to general linear constraints.

Additionally, the viability of the approach should be given a more extensive test by applying the method to a real aircraft and inventory with, consequently, many more variables than are present in the study problems. With this test, some remaining questions (such as the number of starts required to determine all significant local minima for a typical application and the number of cycles required for convergence on a large real problem) could be answered.

Acknowledgments

A portion of this work was supported by the Naval Air Systems Command under Contract N00019-76-C-0160.

References

[1] Ferman, M. A., "A Rapid Method for Flutter Clearance of Aircraft with External Stores," *Theory and Application*, AFFDL-TR-73-74, Vol. 1, Sept. 1973.

[2] Cross, A. K. and Albano, E. A., "Computer Techniques for the Rapid Flutter Clearance of Aircraft Carrying External Stores," AFFDL-TR-72-114, Pt. 1, Feb. 1973.

[3] Gellatly, R. A., Gallagher, R. H., and Luberacki, W. A., "Development of a Procedure for Automated Synthesis of Minimum Weight Structures," AFFDL-TDR-64-141, Oct. 1964.

[4] Gwin, L. B. and Taylor, R. F., "A General Method for Flutter Optimization," *AIAA Journal*, Vol. 11, Dec. 1973, pp. 1613-1617.

[5] Haftka, R. T., "Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements," NASA TN D-7264, 1973.

[6] Stroud, W. J., "Automated Structural Design with Aeroelastic Constraints: A Review and Assessment of the State of the Art," Symposium on Structural Optimization, ASME Winter Annual Meeting, New York, Nov. 1974.

[7] Aoki, M., *Introduction to Optimization Techniques*, MacMillan, New York, 1971.

[8] Davidon, W. C., "Variable Metric Methods for Minimization," Argonne National Laboratory, Ill., Rept. ANL-5990, Nov. 1959.

[9] Murtagh, B. A. and Sargent, R. W. H., "Computational Experience with Quadratically Convergent Minimization Methods," *Computer Journal*, Vol. 13, May 1970, pp. 185-194.

[10] Wilkinson, K. et al., "An Automated Procedure for Flutter and Strength Analysis and Optimization of Aerospace Vehicles," AFFDL-TR-75-137, Vols. I and II, Dec. 1975.

[11] Rudisill, C. S. and Bhatia, K. G., "Optimization of Complex Structures to Satisfy Flutter Requirements," *AIAA Journal*, Vol. 9, Aug. 1971, pp. 1487-1491.

[12] Chipman, R. R. and Malone, J. B., "Study of a Method for Determining Critical Store Configurations for Wing-Store Flutter," Grumman Rept. ADCR-76-1, Sept. 1976.

[13] Giesing, J. P., Kalman, T. P., and Rodden, W. P., "Subsonic Steady and Oscillatory Aerodynamics for Multiple Interfering Wings and Bodies," *Journal of Aircraft*, Vol. 9, Oct. 1972, pp. 693-702.

[14] Hassig, H. J., "An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration," *Journal of Aircraft*, Vol. 8, Nov. 1971, pp. 885-889.