Mode Tracking Issues in Structural Optimization

M. S. Eldred*
University of Michigan, Ann Arbor, Michigan 48109
V. B. Venkayya†
U.S. Army Aeroflightdynamics Directorate, Wright–Patterson Air Force Base, Ohio 45433
and
W. J. Anderson‡
University of Michigan, Ann Arbor, Michigan 48109

Within the context of optimization of the structural dynamics properties of finite element models, methodology is developed for the tracking of eigenpairs through changes in the structural eigenvalue problem. The goal is to eliminate difficulties caused by "mode switching" (i.e., frequency crossing). Out of several candidate methods, two methods for mode tracking are successful. The first method, the higher order eigenpair perturbation algorithm, is based on a perturbation expansion of the eigenproblem. It iteratively computes changes in the eigenpairs due to parameter perturbations with the important feature of maintaining the correspondence between the baseline and perturbed eigenpairs. The second method is a cross-orthogonality check method, which uses mass orthogonality to reestablish correspondence after a standard reanalysis. Modified eigenpair extraction routines (Lanczos, subspace iteration, inverse power) were unsuccessful in tracking modes. Applications of mode tracking technology that are presented are frequency-constrained optimization and optimization with mode shape constraints. Each application procedure is outlined and examples are given. Recommendations are made based on method efficiency and robustness in the example problems.

Nomenclature

b	= design variable
$[C]$	= cross-orthogonality check matrix
c	= weighting factor for the homogeneous solution in the eigenvector perturbation calculation
$[D]$	= singular coefficient matrix in the eigenvector perturbation solution
$\{F\}$	= static pseudoload vector appearing in the eigenvector perturbation calculation
f	= frequency, Hz
$[K]$	= stiffness matrix; symmetric
$[M]$	= mass matrix; symmetric
$\{V\}$	= particular solution for eigenvector perturbation from Nelson's method
Δ	= perturbation symbol denoting exact change from a reference
λ	= eigenvalue; ω^2
ξ	= collection of terms in solution for c
$[\Phi]$	= matrix of $\{\phi\}$ column vectors
$\{\phi\}$	= normal mode shape; normalized with respect to appropriate $[M]$
ω	= circular frequency, rad/s

Subscript

i	= associated with the ith eigenpair

Superscripts

(k)	= optimization iteration number
T	= transpose

0	= baseline values
1	= perturbed values resultant from parameter change, e.g., $()^1 = ()^0 + \Delta()$
*	= user-specified quantity
~	= error from a specified quantity

Introduction

IN modern structural optimization software, the performance of a structural design is optimized with respect to many different performance metrics. In the automated structural optimization system (ASTROS), for example, constraints can be applied on stress, displacements, buckling loads, natural frequencies, and static and dynamic aeroelastic properties. In optimization using free vibration characteristics, modal quantities (frequencies and mode shapes) may be specified in the objective function or in the constraints. For this class of optimization problem, mode tracking is required to facilitate the proper bookkeeping of the specified frequencies and mode shapes.

When optimizing structural dynamic characteristics, specific frequencies and mode shapes must be referenced by a number. In vibration problems, the eigenvalues and eigenvectors are ordered by eigenvalue magnitude. When design variable perturbations are performed, frequencies will drift and mode crossings can occur. If these crossings are not tracked, the objective function and constraint functions can be evaluated using modes that are different from those that were intended. This causes problems both from the design standpoint (the optimum found will not reflect the design goals) and from the mathematical viewpoint (convergence can be destroyed).

Theory of Mode Tracking

Mode tracking algorithms maintain correspondence between baseline and perturbed eigenpairs (sets of frequencies and modes on adjacent iterations) through changes in the parameters of an eigenproblem statement. They may either augment a standard eigenproblem analysis or replace it entirely, depending on the technique used. Mode tracking is applicable whenever repeated eigenproblem analyses are required, such as within an optimization procedure or within an incremental process. The former case of mode tracking within optimization is investigated herein, whereas the latter case of mode tracking within an incremental process is investigated in
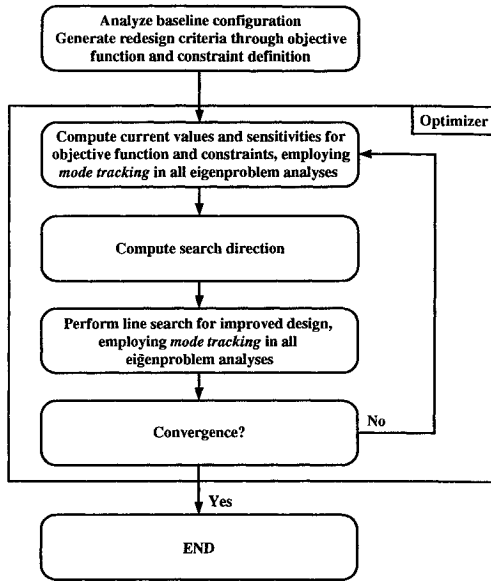
Fig. 1 Nonlinear programming optimization algorithm with mode tracking.

Ref. 1 for the case of incremental aeroelastic analysis by the V-g and p-k methods.

A simple flowchart for optimization with mode tracking is shown in Fig. 1, for the case of a gradient-based optimizer (nonlinear programming). The first step is the generation of a baseline model. After thorough analysis of the baseline model, criteria are developed for desired improvements and are implemented through a design objective and design constraints. Within the optimizer, a nonlinear programming algorithm requires current values and sensitivities for the objective function and constraints to calculate a search direction. For optimization of structural dynamics quantities, an eigenproblem analysis must be performed for this evaluation, and mode tracking must be employed to maintain proper identification of the frequencies and modes of interest through whatever design changes have occurred since the previous eigenproblem analysis. This eigenproblem analysis may occur internally in the case of an integrated optimization system (e.g., ASTROS, MSC/NASTRAN) or externally in the case of a nonintegrated system. Once the search direction has been computed, a line search along this search direction is performed. This generally requires additional objective function and constraint evaluations (an exception is when approximation concepts are used exclusively in the line search), and again mode tracking must be employed in each eigenproblem analysis. Based on the results of the line search, a new estimate of the optimal design variables is calculated, and if convergence has not been obtained, another optimization cycle is performed.

Mode Tracking Methods for Self-Adjoint Eigenvalue Problems

Two successful methods, the higher order eigenpair perturbation algorithm and the cross-orthogonality check method, are proposed for use in mode tracking for self-adjoint problems. One unsuccessful class of methods, the modified eigenpair extraction routine, is briefly discussed.

Higher Order Eigenpair Perturbations

The thrust of the higher order eigenpair perturbation (HOEP) method[2] is to solve for perturbations $\Delta\lambda_i$ and $\{\Delta\phi\}_i$ via an exact perturbation expansion of the standard undamped structural eigenproblem:

$$[K^0]\{\phi^0\}_i = \lambda_i^0[M^0]\{\phi^0\}_i \tag{1}$$

When the system is perturbed from the baseline through mass and stiffness changes, the new statement of dynamic equilibrium is

$$[K^1]\{\phi^1\}_i = \lambda_i^1[M^1]\{\phi^1\}_i \tag{2}$$

or in expanded form

$$([K^0] + [\Delta K])(\{\phi^0\}_i + \{\Delta\phi\}_i)$$
$$= (\lambda_i^0 + \Delta\lambda_i)([M^0] + [\Delta M])(\{\phi^0\}_i + \{\Delta\phi\}_i) \tag{3}$$

All perturbation terms are retained, which leads to coupled equations for $\Delta\lambda_i$ and $\{\Delta\phi\}_i$. These equations must be solved iteratively and will converge to the exact eigenpair perturbations.

The full-order eigenvalue perturbation equation, derived by premultiplying Eq. (3) by $\{\phi^0\}_i^T$ and twice canceling the baseline solution of Eq. (1) (see Ref. 2 for a detailed derivation), is

$$\Delta\lambda_i = \frac{\{\phi^0\}_i^T ([\Delta K] - \lambda_i^0[\Delta M])\{\phi^1\}_i}{\{\phi^0\}_i^T[M^1]\{\phi^1\}_i} \tag{4}$$

where, in the context of a reanalysis phase in an optimization process, the superscript 0 denotes the previous design and the superscript 1 denotes the current design. Unlike a linear eigenvalue sensitivity expression, in which only the baseline eigenvector appears, Eq. (4) shows coupling with the perturbed eigenvector. This equation provides a more accurate estimate of λ_i^1 than the Rayleigh quotient, because the Rayleigh quotient introduces more error for an approximate eigenvector.

The corresponding eigenvector perturbation equation, derived by canceling the baseline solution in Eq. (3) and collecting $\{\Delta\phi\}_i$ terms, is

$$[D^1]_i\{\Delta\phi\}_i = \{F^{nl}\}_i \tag{5}$$

where

$$[D^1]_i \equiv [K^1] - \lambda_i^1[M^1] \tag{6}$$

is singular, and

$$\{F^{nl}\}_i \equiv (\Delta\lambda_i[M^1] + \lambda_i^0[\Delta M] - [\Delta K])\{\phi^0\}_i \tag{7}$$

is a static pseudoload. Equation (5) corresponds to the pathological Fredholm alternative in which the coefficient matrix is singular, yet a nonzero load exists. Such equations cannot be solved in general. This equation is solvable, however, since it is "consistent." For self-adjoint systems, consistency requires that the forcing vector be orthogonal to the null space of the singular coefficient matrix (the forcing vector does not "push" in a singular direction). Thus, for Eq. (5), consistency requires that $\{F^{nl}\}_i^T\{\phi^1\}_i = 0$, since the perturbed eigenvector defines the null space of $[D^1]_i$ ($[D^1]_i\{\phi^1\}_i = \{0\}$ by definition). The null result of $\{F^{nl}\}_i^T\{\phi^1\}_i$ can be shown by substituting Eq. (4) into Eq. (7) and performing the inner product.

The total solution for $\{\Delta\phi\}_i$ is the sum of homogeneous and particular solutions, $\{\phi^1\}_i$ and $\{V\}_i$, respectively. For singular $[D^1]_i$, $\{\phi^1\}_i$ is a homogeneous solution for $\{\Delta\phi\}_i$ in Eq. (5) since $[D^1]_i\{\phi^1\}_i = \{0\}$. The weighting factor c_i is introduced because the scaling of the homogeneous solution is initially indeterminate:

$$\{\Delta\phi\}_i = c_i\{\phi^1\}_i + \{V\}_i \tag{8}$$

Equation (8) must be altered since $\{\phi^1\}_i$ is unknown. Expanding $\{\phi^1\}_i$ and collecting $\{\Delta\phi\}_i$ terms gives

$$\{\Delta\phi\}_i = \frac{c_i}{1 - c_i}\{\phi^0\}_i + \frac{1}{1 - c_i}\{V\}_i \tag{9}$$

Employing equations of mass normalization for the current and projected systems and substituting Eq. (9) for $\{\Delta\phi\}_i$ yields a quadratic equation for c_i. The desired root is the one that minimizes the contribution of the homogeneous solution in Eq. (8) (i.e., the smaller of the two roots):

$$c_i = 1 - \sqrt{1 + \xi_i} \tag{10}$$

where

$$\xi_i = \{\phi^0\}_i^T[\Delta M]\{\phi^0\}_i + 2\{\phi^0\}_i^T[M^1]\{V\}_i + \{V\}_i^T[M^1]\{V\}_i \tag{11}$$
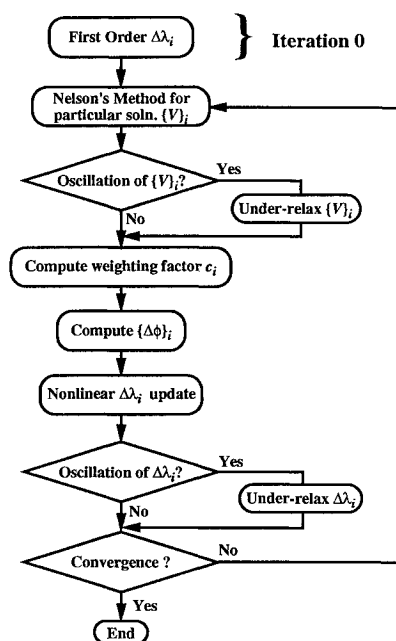
Fig. 2 HOEP algorithm with under-relaxation.

The denominator $(1 - c_i)$ in Eq. (9) can never be zero, since this occurrence would require that $\{V\}_i = -\{\phi^0\}_i$ from Eq. (8). This solution for $\{V\}_i$ is known as the "spurious" solution and is avoided by removal of the singularity with Nelson's method[3] (see Ref. 4 for more details).

Algorithm

The algorithm has evolved considerably since its original publication.[2] It has been greatly simplified, and its convergence characteristics have been improved. The current algorithm flowchart is shown in Fig. 2. Iteration 0 consists of obtaining an initial estimate of the eigenvalue perturbations from a first-order approximation to Eq. (4). The nonlinear iterations are then performed, which consist of solution of the almost singular eigenvector perturbation equation [Eq. (5)] followed by the full-order update for the eigenvalue perturbation [Eq. (4)]. Solution for the eigenvector perturbation requires the computation of the particular solution $\{V\}_i$ of Eq. (5) by Nelson's method,[3] calculation of c_i from Eqs. (10) and (11), and finally solution of Eq. (9) for $\{\Delta\phi\}_i$. These iterations continue until the convergence criterion is satisfied. Nelson's method is successful despite the fact that $[D^1]_i$ is only approximately singular (for a λ_i^1 estimate that has not fully converged), as it removes the singularity regardless of its condition. The important feature for mode tracking is the fact that the computations involve data from the ith eigenpair only, creating a direct correspondence between previous and current modal data and eliminating the problems associated with frequency crossings. The method is computationally intensive but can converge for very large prescribed changes in design variables.[2,4]

Computational expense of the HOEP algorithm is a major concern for large finite element models, as decomposition of a full-order matrix $([D^1]_i)$ is involved each time the Nelson's method step of the HOEP algorithm is performed. As such, alternatives to repeated decomposition, such as use of a Lanczos-based solution in place of Nelson's method (see Ref. 5), are worth investigating in the future.

Relaxation

Relaxation has proved to be useful in accelerating HOEP convergence. If the design changes are large or if the convergence criterion is not strict enough over a series of design cycles, the HOEP iterations may oscillate and converge slowly. These oscillations result from repetitive overprediction of the eigenpair changes. Detection of these oscillations is straightforward and requires monitoring of the sign of the changes in $\Delta\lambda_i$ and of the orientation of the vector changes in $\{V\}_i$ from one HOEP iteration to the next. When oscillation of $\Delta\lambda_i$ or $\{V\}_i$ is detected, underrelaxation is performed.

This entails accepting only a portion (50–100%) of the predicted change in $\Delta\lambda_i$ or $\{V\}_i$. For large oscillations, values closer to 50% work better, and for mild oscillations, values closer to 100% work better. A variable underrelaxation parameter is used in the example problems that assumes values between 0.5 and 1.0 based upon the magnitude of the oscillations.

Repeated Eigenvalues

The subject of repeated eigenvalues has recently received attention in the literature,[6–9] and it is of importance here since Nelson's method will fail when λ_i^1 is repeated. The authors view repeated eigenvalues as a rare occurrence for most complex structures, but structures possessing multiple planes of symmetry, cyclic symmetry, or axisymmetry (which cannot undergo model simplification based on symmetry) certainly merit concern. The computation of the differentiable mode shapes and their derivatives is well documented in the references, and the differences here are that perturbations will be computed instead of derivatives and that the rank deficiency of $[D^1]_i$ occurs for repeated λ_i^1 instead of for repeated λ_i^0 (as it does when calculating eigenvector derivatives). If a baseline eigenvalue is repeated, then the differentiable mode shapes are computed for use as the $\{\phi^0\}_i$ vectors in Eqs. (4) and (7). If a perturbed eigenvalue is repeated, then the additional singularities must be removed from $[D^1]_i$ through additional matrix modifications.[8] Even though perturbed eigenvectors are not uniquely defined for repeated perturbed eigenvalues, the $\{\Delta\phi\}_i$ vectors computed by HOEP are unique when the additional singularities in $[D^1]_i$ have been removed. They, in turn, define the differentiable perturbed eigenvectors for use as the differentiable baseline eigenvectors on the next step of an incremental process. Thus, the computation of differentiable mode shapes would only need to be performed if a repeated condition exists on the first step of an incremental process.

A problem that has not been addressed in the literature is the detection of repeated eigenvalue conditions. If all eigenvalues in a range are extracted at each design step, then monitoring for repeated baseline eigenvalues can be as simple as measuring eigenvalue separation. If, however, there are eigenvalues in the range that are not extracted (such as when tracking only certain eigenpairs), the repeated baseline eigenvalue condition is not trivially determined, and the rank of $[K^0] - \lambda_i^0[M^0]$ must be monitored. Furthermore, the repeated perturbed eigenvalue condition is never trivially determined since the separation of these eigenvalues is not known. For this case, the rank of $[D^1]_i$ must be monitored. One computationally expensive method for monitoring rank deficiency is by singular value decomposition, and more economical methods may be achievable.

Modified Eigenpair Extraction Routines

The block Lanczos, subspace iteration, and inverse power eigenpair extraction routines were identified as having potential for mode tracking. In all cases, the idea was to preserve the baseline eigenpair ordering by using the previous iteration eigenpair data for trial vectors and shift values. In standard usage, random trial vectors are generated, shift values are widely spaced, and the converged eigenpairs are returned in ascending eigenvalue order. This reordering destroys the one-to-one correspondence if mode switching has occurred. The goal of the modifications was to eliminate the reordering by altering the convergence characteristics of the extraction methods. In each case, however, the ability to track modes was not achieved.

Modification of the block Lanczos method[10,11] was quickly eliminated from consideration because the iterates in the Lanczos recursion, the Lanczos vectors, are not directly related to the eigenvectors. Therefore, the use of baseline eigenvectors as trial vectors is not appropriate.

The subspace iteration algorithm is an eigenpair extraction routine that, unlike the Lanczos method, iterates directly on eigenvectors.[12] The modifications fail in the task of mode tracking because convergence in the subspace iteration method occurs according to eigenvalue magnitude and not according to eigenvector similarity to trial vectors. Although convergence is improved when using the baseline eigenvectors since the starting subspace is an excellent approximation to the least-dominant (i.e., converged) subspace, convergence

still occurs in eigenvalue order due to the minimization of the Rayleigh quotient inherent in the method. Shifting was also ruled out since it would be inefficient to shift on each baseline eigenvalue and, more importantly, since the procedure would not be foolproof. It is easy to envision scenarios where the closest eigenvalue to a shift is not the correct eigenvalue. This could occur for closely spaced eigenvalues or for frequency crossings, the cases of most concern in this paper.

Modifications to the inverse power method with shifting and sweeping would be unsuccessful for the same reasons that subspace iteration modifications were unsuccessful; that is, convergence occurs on eigenvalue magnitude rather than trial vector similarity, and shifting is neither fail safe nor efficient.

Thus, although some of the modified eigenpair extraction methods could be used for fast reanalysis due to their improved convergence speed, none of the methods exhibit a reliable ability to track modes.

Cross-Orthogonality Check (CORC)

This method, proposed by Gibson,[13] performs a mass orthogonality check after reanalysis. The same basic method was recently proposed by Ting,[14] who recognizes the equivalence of the modal assurance criterion (MAC) used in ground vibration tests (GVTs). The orthogonality information is held in the following mass triple product,

$$[C] = \left[\Phi^{(k-1)}\right]^T \left[M^{(k)}\right]\left[\Phi^{(k)}\right] \tag{12}$$

where k and $k-1$ are the current and previous iterations, respectively. If the $[C]$ matrix is diagonally dominant, then no mode switching has occurred; and if the matrix is not diagonally dominant, the locations of the dominant values can be used to recorrelate the current iteration modes.

It is important to monitor the assurance with which the correlations are made. This is accomplished through the computation of "corruption indices." A corruption index for a mode in the CORC method is the largest magnitude value in the column of $[C]$ different from the correlated value (with a reference value of 1). For example, if a column of $[C]$ has first and second largest component magnitudes of 1.2 and 0.3, then the value of 1.2 correlates the mode and the corruption index for that mode is $0.3/1.2 = 0.25$. The maximum corruption index shows the largest corruption index over the number of correlated modes. The assurance of proper mode correlation decreases as the size of the corruption indices increases. Typically, a corruption value of 0.5 or higher merits concern.

The method does not attempt to directly track modes but rather tries to reestablish correspondence after the changes in design and mode switches have occurred. Although attractive due to its simplicity, the method is unattractive in that the standard reanalysis does not make use of available baseline information and, in a sense, starts from scratch (exception: use of fast reanalysis techniques).

The use of mass matrix orthogonality information is not new, and considerable work in this area appears in the system identification and model correlation literature. Among these, Refs. 15 and 16 are concerned with enhancing the orthogonality of a set of measured modes from a GVT. Though the applications are different, these methods share a common mathematical basis with the cross-orthogonality check method.

Applications

Optimization with Frequency Constraints

Problems of structural optimization subject to frequency constraints can be formulated as follows:
Minimize

$$f(\{b\}) \tag{13}$$

subject to

$$g_j(\{b\}) = \lambda_i - \lambda_i^* \geq 0$$

where it is understood that an upper-bound frequency constraint differs only in sign from the lower-bound constraint shown. The
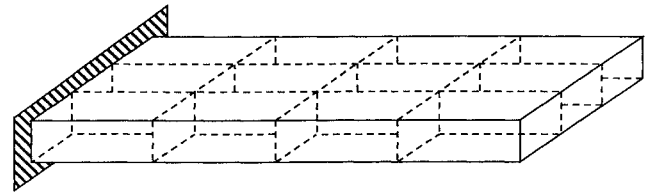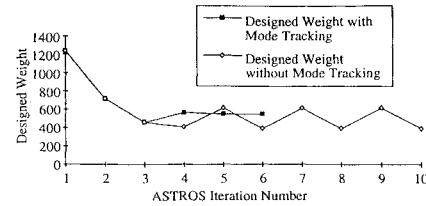

Fig. 3 Simple cantilevered wing box.

Fig. 4 Simple wing box weight minimization with a frequency constraint.

linear sensitivity of this frequency constraint to the kth design variable is

$$\frac{\partial g_j}{\partial b_k} = \frac{\partial \lambda_i}{\partial b_k} = \frac{\{\phi\}_i^T ([\partial K/\partial b_k] - \lambda_i[\partial M/\partial b_k])\{\phi\}_i}{\{\phi\}_i^T [M]\{\phi\}_i} \tag{14}$$

where the denominator is unity in the case of mass normalized eigenvectors.

Since the eigenvectors do not appear in the preceding problem formulation [Eq. (13)], it is tempting for the analyst to ignore the eigenvectors altogether. Even with modest design changes, however, this practice can fail due to the possibility of mode switches. When this possibility occurs, the optimization iteration history can exhibit a "jump" since the constraint sensitivity is being calculated for a vastly different mode shape. The iterations may even oscillate unpredictably if the mode switching is recurrent. Thus, even if frequency constraints are the only concern, mode tracking is an important technology.

Example 1: Simple Cantilevered Wing Box

A simple cantilevered wing box (Fig. 3) is created here for use in optimization with vibration constraints. The ASTROS multidisciplinary optimization code is being used with matrix programming oriented language (MAPOL) coding and FORTRAN modules performing all nonstandard tasks. The simple wing box is to undergo weight minimization subject to frequency constraints. The third and fourth frequencies are closely spaced in the original design and have exhibited a tendency to switch under design changes.

The first optimization procedure is the unmodified ASTROS procedure that makes use of a standard reanalysis. When a lower-bound constraint is placed on f_3 ($f_3 \geq 110$ Hz), optimization without mode tracking exhibits large oscillations (see Fig. 4). The modes first switch on iteration 3, causing the constraint sensitivity to be calculated using the wrong mode shape and resulting in an erroneous search direction for the iteration 4 design. Since ASTROS uses approximation concepts exclusively in the line search step, the poor search direction is not detected with eigenproblem analyses, and the new design point is not an improvement. The modes have switched back in the iteration 4 design and the optimization begins its nonconverging oscillations. The design never achieves feasibility, i.e., the frequency constraint is never satisfied. When mode tracking by the HOEP or CORC algorithms is inserted in place of the reanalysis routine, the mode switch on iteration 3 is properly tracked and the correct sensitivities are calculated. Feasibility is achieved on iteration 4 and Fig. 4 shows convergence to the optimal design in 6 iterations.

Mode tracking handles frequency crossings by enabling constraint enforcement on specific modes that are ordered for all iterations according to their positions in the initial design. For the current case, mode 3 in the initial design satisfies the 110-Hz constraint in the final design, even though its frequency has become the fourth lowest. If the design objective in this example had been to

Table 1   Performance of CORC and HOEP for simple
                cantilevered wing box

ASTROS iter. no.	Givens + CORC		HOEP	
	CPU	Max. corr.	CPU	Iters.
2	2.6	0.0758	3.3	2
3	2.6	0.128	3.2	2
4	2.6	0.163	4.3	3
5	2.6	0.0239	3.2	2
6	2.6	0.00121	2.0	1

raise the third lowest frequency regardless of the mode to which it belongs, then that goal would not have been met due to improper problem formulation. This illustrates a very important point. It is essential to clearly distinguish the difference between the design goal of controlling the frequency of a specific mode, as in this example, and the goal of controlling the nth ordered frequency regardless of the mode to which it belongs. Examples of the latter case are generally simpler and might involve avoidance of a resonance condition in some frequency band where the modal character of the resonance is unimportant. Examples of the former case involve more sophisticated design goals, such as in test and analysis model reconciliation, where analytical modal data are driven towards experimental modal data and the modal character of frequencies is always important. An example of this importance is the fact that driving together the analytical and experimental frequencies of different modes would be meaningless. Mode tracking provides an automatic means to achieve the goal of precise modal control, which otherwise could not be achieved, whereas the case of ordered frequency control can be handled with or without mode tracking so long as an appropriate problem formulation is used (e.g., to smooth the oscillation in Fig. 4, apply identical constraints on each of the switching modes and either use exact analyses in the line search or, if employing the approximation concepts approach, use a more limit decreasing strategy). In general, mode tracking provides more precise control of modal data for the cases where more precise control is needed.

Both HOEP and CORC are robust enough to handle the mode switching in this example problem. In general, the difficulty of an optimization mode tracking task is dependent upon the structure involved and the optimization move limits that are used. The optimization move limits constrain how large a move in the design space the line search can make and therefore largely determine the magnitude of the changes in mode shape that can occur. The mode tracking task becomes more difficult as larger changes in mode shape are allowed from one iteration to the next. Thus, if one is willing to pay the penalty of higher optimization iterations to reach the structural optimum, one can generally decrease the mode tracking difficulty by decreasing the move limits. ASTROS uses a factor of 2.0 design variable move limit as its default. This means that any design variable can, at most, halve or double during a design change. This default was employed in this example, and both mode tracking methods performed well for this relatively generous move limit.

The question of relative efficiency depends on several variables, including the method of reanalysis used and the range of modes extracted before CORC and the number of modes tracked and the convergence tolerance used in HOEP. The reanalysis before CORC must find eigenpairs in a range of frequency to ensure that the modes to be tracked are extracted. HOEP, on the other hand, has the advantage of tracking only the constrained eigenpairs. In this example, CORC augments a Givens method extraction that computes eigenvectors for frequencies less than 200 Hz (7 eigenvectors), and HOEP tracks eigenpair 3 with a convergence tolerance of 0.01% on λ_3^1 change (HOEP convergence is indicated when the change in the λ_3^1 estimate between iterations is less than 0.01%). CPU time is measured in seconds for a DEC 5000 workstation. Table 1 shows the performance data for the five optimization iterations following the baseline analysis.

The CPU time for the Givens reanalysis alone was 2.2 s per analysis, and so the actual orthogonality check is inexpensive (0.4 s). The largest corruption value for CORC was 0.163, an acceptable value. HOEP converged very quickly but was 23% more computationally expensive on average. These data show that the mode tracking task
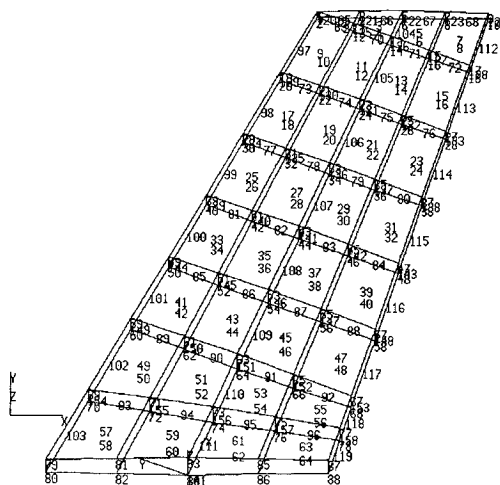

Fig. 5   Intermediate complexity wing.
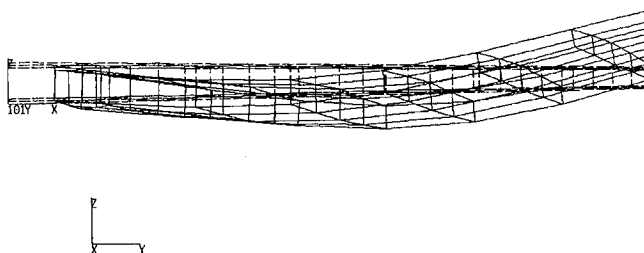

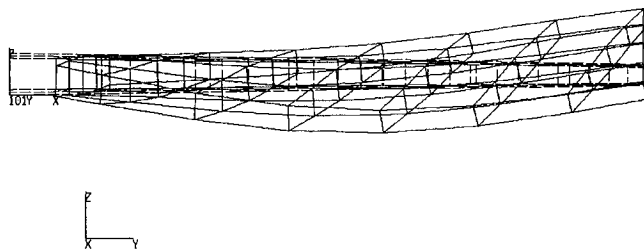Fig. 6   ICW baseline mode shape 2.

Fig. 7   ICW baseline mode shape 3.

was not difficult for this example. The reason for this is that the mode shapes are well differentiated in this example, since the box cross section is doubly symmetric and the design variables are linked so as to retain one plane of symmetry. In example 2, such symmetries do not exist and the mode shapes are less well differentiated.

Example 2: Intermediate Complexity Wing

The intermediate complexity wing (ICW) shown in Fig. 5 is a well-known benchmark optimization model.[17,18] This wing is to undergo weight minimization subject to a constraint on its second frequency ($f_2 \geq 8$ Hz). The mode tracking task is more difficult than in example 1, because the mode shapes are less well differentiated. Figures 6 and 7 show the physical similarity between baseline modes 2 and 3 (second bending and first torsion, respectively).

Figure 8 shows the history for optimization without mode tracking and for optimization with HOEP and CORC mode tracking. As for example 1, the default factor of 2.0 design variable move limit was employed. Once again, a mode switch causes optimization without mode tracking to exhibit oscillations. For optimization with mode tracking, the mode switch on iteration 4 is properly tracked and the correct sensitivities are calculated. Figure 8 shows convergence to the optimal design in 10 or 12 iterations. The histories for HOEP and CORC were not identical because the optimizer was sensitive to very small differences in mode shape for this example. That is, very small differences in the modes calculated by Givens reanalysis and by perturbation expansion caused noticeable differences in the design path taken by the optimizer. This result should not be interpreted

Table 2 Performance of CORC and HOEP for ICW

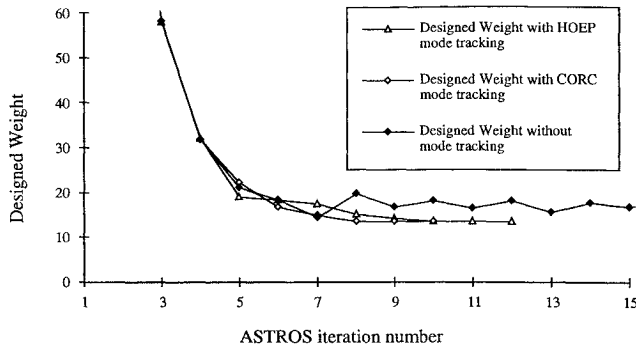| ASTROS iter. no.	Givens + CORC		HOEP	
	CPU	Max. corr.	CPU	Iters.
2	23.3	0.956	42.3	7
3	23.0	0.254	28.0	5
4	23.2	0.299	28.1	5
5	23.3	0.591	38.8	7
6	23.2	0.286	33.5	6
7	23.6	0.129	39.5	7
8	23.5	0.183	68.1	13
9	23.6	0.0342	59.8	11
10	23.7	1.4e−13	79.1	13
11	——	——	45.8	8
12	——	——	7.3	1

Fig. 8 ICW weight minimization with a frequency constraint.

as an advantage of one mode tracking method over the other, since the observed variations in design paths did not consistently favor either method.

Performance data for the two mode tracking methods are shown in Table 2. To minimize the differences in mode shape between HOEP and CORC, the convergence tolerance for HOEP was kept strict: less than $10^{-6}\%$ change in λ_2^1 between iterations indicates convergence. CORC augments a Givens reanalysis that computes eigenvectors for frequencies less than 25 Hz (seven, eight, or nine eigenvectors, depending on the iteration).

It is evident from the higher CORC corruption indices and higher HOEP iterations to convergence that this mode tracking task was more difficult than in example 1. The maximum corruption index of 0.956 on ASTROS iteration 2 is unacceptably high, but it was not for the correlation of mode 2 and thus was not critical. However, the maximum corruption index of 0.591 on ASTROS iteration 5 did occur for the correlation of mode 2 and therefore shows near failure of CORC for this design task. The higher number of HOEP iterations needed to satisfy the convergence criterion was due partly to the stricter criterion and partly to the increased difficulty of the mode tracking task. As a result of the higher number of iterations needed, HOEP is noticeably more expensive than CORC in this example. HOEP, however, did not exhibit any convergence difficulties and therefore appears to be more robust than CORC for this difficult mode tracking task.

Optimization with Mode Shape Constraints

An extension of the frequency-constrained optimization problem is to include constraints on the mode shapes. This new capability would enable an analyst to constrain a mode to have some specified shape. This might involve controlling nodal lines to minimize vibration in a region of the structure or minimizing discrepancies with experimentally measured mode shapes. Although constraints of this type can be formulated in numerous ways due to the vector nature of a mode shape, the approach employed here will involve vector norms. If, for example, it is desired to keep the ith mode shape close to some specified shape, then the optimization problem can be formulated as follows:

Minimize

$$f(\{b\}) \tag{15}$$

subject to

$$g_j = \mu_i^* - \mu_i \geq 0$$

where

$$\mu_i = \|\{\widetilde{\Delta\phi}\}_i\|$$

and

$$\{\widetilde{\Delta\phi}\}_i = \{\phi\}_i - \{\phi^*\}_i$$

Here, $\{\phi\}_i$ is the current mode shape and $\{\phi^*\}_i$ is the specified mode shape. The μ_i^* is user specified to denote how close the current eigenvector must be to the prescribed eigenvector. To lead to a simpler sensitivity formula, the constraint can be reformulated with the square of the vector norm:

$$g_j = (\mu_i^*)^2 - \|\{\widetilde{\Delta\phi}\}_i\|^2 \geq 0 \tag{16}$$

or equivalently

$$g_j = (\mu_i^*)^2 - \{\widetilde{\Delta\phi}\}_i^T \{\widetilde{\Delta\phi}\}_i \geq 0 \tag{17}$$

This does not alter the nature of the constraint. The sensitivity of this new constraint to a design variable is

$$\frac{\partial g_j}{\partial b} = -2\{\widetilde{\Delta\phi}\}_i^T \frac{\partial \{\phi\}_i}{\partial b} \tag{18}$$

where the fact that

$$\frac{\partial \{\widetilde{\Delta\phi}\}_i}{\partial b} = \frac{\partial \{\phi\}_i}{\partial b}$$

has been used, since

$$\frac{\partial \{\phi^*\}_i}{\partial b} = 0$$

In an alternate approach, the current mass matrix could be used to weight the different vector components in the inner product of Eq. (17). This adds a term involving $[\partial M/\partial b]$ to the sensitivity expression but might be desirable depending on the application.

Implementing the sensitivity in Eq. (18) requires calculation of the eigenvector derivative $\partial \{\phi\}_i/\partial b$ (by Nelson's method[3] or other methods). The $\{\widetilde{\Delta\phi}\}_i$ vector, the difference of the current and prescribed eigenvectors, should not be confused with the $\{\Delta\phi\}_i$ vector, the eigenvector perturbation in the HOEP algorithm.

This constraint formulation is very general since any portion of an eigenvector can be prescribed. If only certain degrees of freedom are to be constrained, then only those freedoms appear in the vector calculations of Eqs. (17) and (18). Common choices for the prescribed eigenvector would include a mode shape from the initial design (enforcing minimum change), an experimental mode shape (enforcing correlation), or a region of zero vibration (enforcing a nodal line).

Once again, mode tracking will be very important for successful optimization, since the vector norm constraint formulation will be very susceptible to mode switches. The same convergence problems could occur as in the frequency-constrained case if mode switches are not properly tracked.

The augmentation or replacement of a standard eigenproblem analysis routine with a mode tracking algorithm is entirely separate from the prescription of vibration constraints, and once this change is made, all types of vibration constraints can be imposed without fear of mode switching difficulties. Strictly from a mode tracking point of view, then, optimization with mode shape constraints presents no new challenges over those of frequency-constrained optimization. The new challenges with mode shape constraints surface in the areas of 1) code modification for constraint inclusion, 2) development of practical methods for user specification of mode shapes with a large number of degrees of freedom, and 3) development of conventions that allow for meaningful comparison of eigenvectors from different optimization iterations. Because of the first challenge, specifically a large developmental overhead with ASTROS, examples of optimization with mode shape constraints have not been performed. Instead, the following example explores the third challenge through the new concept of iteration invariance.

Testing has been conducted on the simple cantilevered wing box (Fig. 3) to define "iteration-invariant" normalization and sign convention methods. The goal is to enable the vector norm calculations of Eqs. (17) and (18) to measure differences in mode shape only, separate from differences in sign or normalization. To accomplish this, both the current eigenvectors and the prescribed eigenvectors ($\{\phi\}_i$ and $\{\phi^*\}_i$) must reflect the same sign and normalization convention for all optimization iterations.

Mass normalization of all eigenvectors to the mass matrix of the initial design has been identified as the best invariant normalization method. This normalization deletes changes in scaling from the vector norms. Care must be taken in calculating the eigenvalue sensitivities from Eq. (14), however, since the scaling of $\{\phi\}_i$ affects the sensitivity calculation and the subsequent search direction. Maximum component normalization can also be rendered iteration invariant by setting the same component to unity throughout the optimization iterations (even if the maximum component location changes). This normalization may be needed when use of a mass matrix is inconvenient (e.g., for experimental mode shapes from modal testing), but it lends itself to certain potential ill-conditioning that mass normalization avoids.

The best sign convention method involves making the maximum magnitude baseline eigenvector component greater than zero and retaining the same component as positive in all subsequent eigenvectors. Through this use of the baseline eigenvectors as a reference, the sign convention is consistent for all iterations. A separate sign convention enforcement is not needed if the iteration-invariant maximum component normalization is used.

For an ASTROS weight minimization subject to a fundamental frequency constraint without mode tracking, Figs. 9 and 10 show a before and after view of iteration invariance. Vector norms of eigenvector change measured from the initial design are shown for the seven lowest modes of the cantilever wing box. In Fig. 9, changes in sign and normalization obscure the changes in mode shape. Figure 10 shows the effect of using an invariant sign convention and normalization (note plot scale change). Pure changes in mode shape are plotted, and a mode switch has become evident through the relatively large vector norms for the switched modes.

This concept of iteration invariance is easily extended to nonself-adjoint eigenvalue problems, e.g., for aeroelastic vibratory modes,[1] which will be important if constraining these modes during optimization for flutter or divergence characteristics [using a constraint similar to Eq. (17)] or if measuring changes in vibratory modes between the parameter increments of an aeroelastic analysis.

Conclusions

Mode tracking techniques have been developed and applied to problems in optimization with free vibration constraints. These techniques are important bookkeeping tools that allow the analyst to maintain proper association of modal data, thereby avoiding confusion caused by mode switching. In addition, these techniques allow for more sophisticated design goals involving precise modal control, such as in test and analysis model reconciliation, which could not otherwise be automatically handled in the presence of frequency crossings.

In optimization with frequency constraints, higher order eigenpair perturbations (HOEP) and the cross-orthogonality check (CORC) have both been shown to be effective in eliminating convergence problems caused by mode switching. For optimization with mode shape constraints, the concept of iteration invariance has been shown to be an important issue that allows for the measurement of pure changes in mode shape, separate from changes in normalization or sign convention.

Factor of 2 design variable move limits (halve or double) have been shown to be acceptable, even for difficult mode tracking tasks (although near failure of CORC is observed for these move limits in the ICW example). Relative efficiency of the methods depends on several variables, including the method of reanalysis used and the range of modes extracted before CORC and the number of modes tracked and the convergence tolerance used in HOEP. The CORC reanalysis must find all eigenpairs in a range of frequency to ensure that the modes to be tracked are extracted. HOEP, on the other hand, has the advantage of tracking only the constrained eigenpairs. HOEP is shown to be more robust, although more computationally expensive, than CORC in the example problems.

If the problem requiring mode tracking technology is fairly well behaved, then the orthogonality check method may be preferable due to its simplicity. If, however, the problem is more difficult and large mode changes are possible, then the eigenpair perturbation method is recommended due to its robustness for optimization with free vibration constraints. A hybrid method using CORC for recorrelation that activates HOEP in times of high CORC corruption could potentially exploit the different strengths of the two methods and maximize overall robustness and efficiency.

Fig. 9 Before iteration-invariance.

Fig. 10 After iteration-invariance.

References

[1]Eldred, M. S., Venkayya, V. B., and Anderson, W. J., "New Mode Tracking Methods in Aeroelastic Analysis," *AIAA Journal*, Vol. 33, No. 7, 1995, pp. 1292–1299.

[2]Eldred, M. S., Lerner, P. B., and Anderson, W. J., "Higher Order Eigenpair Perturbations," *AIAA Journal*, Vol. 30, No. 7, 1992, pp. 1870–1876.

[3]Nelson, R. B., "Simplified Calculation of Eigenvector Derivatives," *AIAA Journal*, Vol. 14, No. 9, 1976, pp. 1201–1205.

[4]Eldred, M. S., "Full Order Eigenpair Perturbations with Mode Tracking Applications in Aeroelasticity and Optimization," Ph.D. Dissertation, Dept. of Aerospace Engineering, Univ. of Michigan, Ann Arbor, MI, 1993.

[5]Zhang, L., He, B., and Yao, M., "Sensitivity Analysis Based on Lanczos Approach in Structural Dynamic Math Model Updating," *Proceedings of the 11th International Modal Analysis Conference (IMAC)*, Society for Experimental Mechanics, Inc., Bethel, CT, 1993, pp. 350–355.

[6]Ojalvo, I. U., "Efficient Computation of Modal Sensitivities for Systems with Repeated Frequencies," *AIAA Journal*, Vol. 26, No. 3, 1988, pp. 361–366.

[7]Mills-Curran, W. C., "Calculation of Eigenvector Derivatives for Structures with Repeated Eigenvalues," *AIAA Journal*, Vol. 26, No. 7, 1988, pp. 867–871.

[8]Dailey, R. L., "Eigenvector Derivatives with Repeated Eigenvalues," *AIAA Journal*, Vol. 27, No. 4, 1989, pp. 486–491.

[9]Hou, G. J. W., and Kenny, S. P., "Eigenvalue and Eigenvector Approximate Analysis for Repeated Eigenvalue Problems," *AIAA Journal*, Vol. 30, No. 9, 1992, pp. 2317–2324.
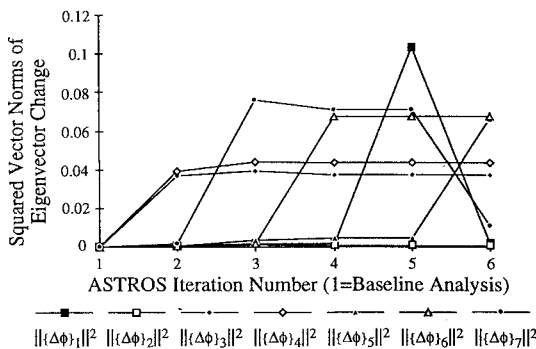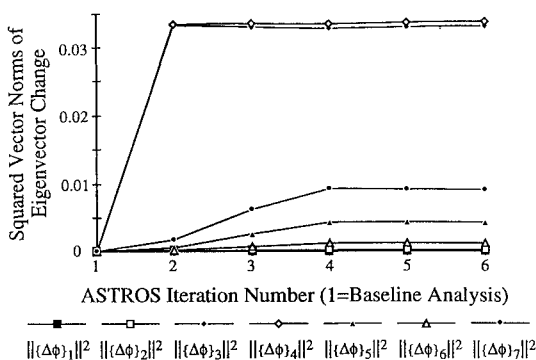
[10]Tischler, V. A., and Venkayya, V. B., "Evaluation of Eigenvalue Routines for Large Scale Applications," *Journal of Sound and Vibration*, Vol. 1, No. 3, 1994, pp. 201–216.

[11]Komzsik, L. (ed)., "MSC/NASTRAN Handbook for Numerical Methods, MSC/NASTRAN Version 66," The MacNeal–Schwendler Corp., Los Angeles, 1990, Sec. 4.4, pp. 1–14.

[12]Bathe, K. J., and Wilson, E. L., *Numerical Methods in Finite Element Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1976.

[13]Gibson, W., "ASTROS-ID: Software for System Identification using Mathematical Programming," Wright Lab. TR WL-TR-92-3100, Wright–Patterson AFB, OH, Sept. 1992.

[14]Ting, T., Chen, T. L. C., and Twomey, W. J., "Automated Mode Tracking Strategy," *AIAA Journal*, Vol. 33, No. 1, 1995, pp. 183–185.

[15]McGrew, J., "Orthogonalization of Measured Modes and Calculation of Influence Coefficients," *AIAA Journal*, Vol. 7, No. 4, 1969, pp. 774–776.

[16]Targoff, W. P., "Orthogonality Check and Correction of Measured Modes," *AIAA Journal*, Vol. 14, No. 2, 1976, pp. 164–167.

[17]Venkayya, V. B., Tischler, V. A., and Pitrof, S. M., "Benchmarking in Structural Optimization," 4th Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 92-4784, Sept. 1992.

[18]Johnson, E. H., and Neill, D. J., "Automated STRuctural Optimization System (ASTROS), Vol. III, Applications Manual," Air Force Wright Aeronautical Lab. TR AFWAL-TR-88-3028, Wright–Patterson AFB, OH, Dec. 1988, pp. 133–182.