Journal of Sound and Vibration (1997) **199**(3), 463–471

FAST EIGENVALUE SENSITIVITY CALCULATIONS FOR SPECIAL STRUCTURES OF SYSTEM MATRIX DERIVATIVES

M. A. El-Kady and A. A. Al-Ohaly

College of Engineering, King Saud University, P.O. Box 800, *Riyadh* 11421, *Saudi Arabia*

(*Received* 12 *February* 1996, *and in final form* 1 *July* 1996)

In this paper a possible application is presented of a general rank-1 matrix formula to the eigenvalue sensitivity evaluation which reduces the sensitivity expressions to elegant, very fast and recursive formulas with substantial savings in computer resources. The rank-1 matrix formula allows for re-arranging terms in multi-product forms involving vectors and matrices. The formula is applicable to rank-1 matrices of special structures which may constitute derivatives of the system state matrix with respect to parameters of interest. In such cases, the use of the rank-1 formula yields exact non-approximate solutions which are identical to those obtained by other conventional formulas. The applicability of the rank-1 formula is believed to cover a wide variety of practical engineering systems pertaining to sound and vibration.

1. INTRODUCTION

The dynamic behaviour of engineering systems, which is a key factor in their design and operation, is closely related to the eigenvalues of the so-called system matrix often referred to in modern control theory to indicate the free or natural response of the system. Eigenvalue sensitivity analysis has, to a great extent, been developed over the years in both theory and applications [1–6]. At present, eigenvalue sensitivity analysis is regarded as an important discipline in systems design and control. As many practical engineering systems today are large-scale in nature, efficient computation of eigenvalue sensitivities with respect to various operating and design parameters is a key requirement in the analysis [7, 8]. Much faster computations and reduced memory requirements can be attained if the special characteristics and structure pertaining to the system under study are utilized.

While working extensively on the problem of eigenvalue sensitivity evaluation, which involves an ordered sequence of product terms of vectors and matrices, the authors of this paper have noticed that, under certain conditions, the result would not change even with some terms in the product sequence exchanging their positions [9]. Such simple manipulation leads to a remarkable reduction in mathematical expressions when the derivatives of the system matrix (with respect to parameters of interest) have special structures, the rank of which equals 1. In this case, the sensitivity expressions reduce to more compact, elegant and recursive schemes leading to much faster computations. Furthermore, those special matrix structures, to which the rank-1 matrix formula is applicable, are believed to be very common in many practical engineering applications [2, 4, 8]. In such applications, only very few elements of the system matrix would depend on a given parameter that is likely to change in practice. In other words, for each sensitivity

0022–460X/97/030463 + 09 $25.00/0/sv960647

parameter of interest, the first and higher order derivatives of the system matrix would be extremely sparse with only few non-zero elements and would, in most cases, constitute special rank-1 forms.

In this paper the authors' findings regarding the rank-1 matrix exchange formula are summarized and one of its powerful applications to eigenvalue sensitivity evaluation, resulting in substantial savings in computations, is illustrated. A wide range of applications could benefit from the developments reported in the paper, including the design and operation of mechanical structures and control schemes, stress analysis and dynamic stability of rotating electrical machines.

2. THEORETICAL INVESTIGATION

Consider an $(n \times m)$ complex (or real) matrix $\mathbf{C}$ (i.e., $\mathbf{C}$ has n rows and m columns) and another $(n \times l)$ complex (or real) matrix $\mathbf{D}$, where both $\mathbf{C}$ and $\mathbf{D}$ are rank-1 matrices which can be expressed in the forms,

$$\mathbf{C} = \mathbf{z}\mathbf{x}^{\mathrm{T}} \qquad \text{and} \qquad \mathbf{D} = \mathbf{z}\mathbf{y}^{\mathrm{T}}, \tag{1a, b}$$

with $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ being $(m \times 1)$, $(l \times 1)$ and $(n \times 1)$ complex (or real) vectors, respectively, and the superscript T denotes transposition. Then, the following theorem is introduced [9].

Theorem. The vector-matrix product equation

$$(\mathbf{h}^{\mathrm{T}}\mathbf{C}\mathbf{g})(\mathbf{q}^{\mathrm{T}}\mathbf{D}\mathbf{p}) = (\mathbf{q}^{\mathrm{T}}\mathbf{C}\mathbf{g})(\mathbf{h}^{\mathrm{T}}\mathbf{D}\mathbf{p}), \tag{2}$$

where $\mathbf{C}$ and $\mathbf{D}$ are given by equation (1), holds true for any $(n \times 1)$, $(m \times 1)$, $(n \times 1)$, $(l \times 1)$ complex (or real) vectors $\mathbf{h}, \mathbf{g}, \mathbf{q}$ and $\mathbf{p}$, respectively.

Proof. The left-hand side of equation (2) is written, by using equations (1), as

$$\mathbf{LHS} = (\mathbf{h}^{\mathrm{T}}\mathbf{z})(\mathbf{x}^{\mathrm{T}}\mathbf{g})(\mathbf{q}^{\mathrm{T}}\mathbf{z})(\mathbf{y}^{\mathrm{T}}\mathbf{p}).$$

By exchanging the first and third (scalar) terms,

$$\begin{aligned}\mathbf{LHS} &= (\mathbf{q}^{\mathrm{T}}\mathbf{z})(\mathbf{x}^{\mathrm{T}}\mathbf{g})(\mathbf{h}^{\mathrm{T}}\mathbf{z})(\mathbf{y}^{\mathrm{T}}\mathbf{p}) = (\mathbf{q}^{\mathrm{T}}(\mathbf{z}\mathbf{x}^{\mathrm{T}})\mathbf{g})(\mathbf{h}^{\mathrm{T}}(\mathbf{z}\mathbf{y}^{\mathrm{T}})\mathbf{p}) \\ &= (\mathbf{q}^{\mathrm{T}}\mathbf{C}\mathbf{g})(\mathbf{h}^{\mathrm{T}}\mathbf{D}\mathbf{p}) = \mathbf{RHS} \qquad \square\end{aligned}$$

Discussion

1. It is apparent that the real value of the above theorem lies in the relative significance of the product terms $(\mathbf{q}^{\mathrm{T}}\mathbf{C}\mathbf{g})$ and $(\mathbf{h}^{\mathrm{T}}\mathbf{D}\mathbf{p})$ in comparison with $(\mathbf{h}^{\mathrm{T}}\mathbf{C}\mathbf{g})$ and $(\mathbf{q}^{\mathrm{T}}\mathbf{D}\mathbf{p})$in equation (2). This, of course would depend on the particular application of interest as well as the practical meaning of the individual product terms, as will be discussed in section 3.
2. It is also noted that the above theorem is valid for any combination of real and/or complex matrices and vectors as long as the proper dimensions are observed.
3. A trivial case of the theorem is obtained when any of the vectors or matrices in equation (2) is zero. On the other hand, a special case of interest would result in when $\mathbf{C}$ and $\mathbf{D}$ are equal, in which case both matrices must be square.
4. The requirement that the matrices $\mathbf{C}$ and $\mathbf{D}$ must be rank-1 and have the form (1) may seem restrictive to the applicability of the theorem. However, as will be seen in the following section, practical applications do exist which contain matrices of the form (1).

3. APPLICATION TO EIGENVALUE SENSITIVITY

One application is presented here in which the rank-1 matrix formula described in the previous section would prove to be very useful. The application involves the calculation of the first and higher order sensitivities of eigenvalues of a real matrix with respect to variations in some parameters which define the elements of the matrix. In practice, the real matrix is often called the system or state matrix and the parameters usually constitute design and/or operation variables in the engineering system on which the elements of the system matrix would depend. The dynamic behaviour of many engineering systems which experience variations in design or operating conditions is closely linked to the eigenvalues of the system matrix. Therefore, the sensitivities of the eigenvalues (system modes) with respect to variations in system parameters are extremely important in assessing the effect of such parameter variations on the overall dynamic performance of the system.

In many practical engineering systems, only very few elements of the system matrix would be affected by changes in a system parameter of interest. This is usually true for most system parameters which are likely to vary during actual system operation or as part of the design and modelling processes. Consequently, the derivatives of the system matrix with respect to such parameters are, in fact, very sparse and, in most cases, form rank-1 matrices. This special characteristic can be exploited, by using the rank-1 matrix formula of this paper, to reduce the eigenvalue sensitivity formulas to much faster, recursive expressions.

3.1. BASIC DERIVATIONS

Consider an $(n \times n)$ system matrix $\mathbf{A}$ with eigenvalues λ_i, $i = 1, \ldots, n$, arranged in a column vector $\boldsymbol{\lambda}$. The eigenvalues, which are assumed to be distinct are related to the corresponding eigenvectors $\mathbf{u}_i$, $i = 1, \ldots, n$, of $\mathbf{A}$ by the equations

$$\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{u}_i, \qquad i = 1, \ldots, n. \tag{3}$$

Similarly, for the transpose of $\mathbf{A}$, one can write [1]

$$\mathbf{A}^{\mathrm{T}}\mathbf{v}_j = \lambda_j \mathbf{v}_j, \qquad j = 1, \ldots, n. \tag{4}$$

It is to be noted that $\mathbf{u}_i$ and $\mathbf{v}_j$ are orthogonal and can be scaled such that

$$\mathbf{v}_i^{\mathrm{T}}\mathbf{u}_i = 1, \qquad \text{and} \qquad \mathbf{v}_j^{\mathrm{T}}\mathbf{u}_i = 0 \quad \text{for } j \neq i. \tag{5a, b}$$

Now, differentiating (3) with respect to a parameter ζ of interest, one obtains

$$\dot{\mathbf{A}}\mathbf{u}_i + \mathbf{A}\dot{\mathbf{u}}_i = \dot{\lambda}_i \mathbf{u}_i + \lambda_i \dot{\mathbf{u}}_i, \tag{6}$$

where $\dot{\mathbf{A}} = (\partial \mathbf{A}/\partial \zeta)$, $\dot{\mathbf{u}}_i = (\partial \mathbf{u}_i/\partial \zeta)$ and $\dot{\lambda}_i = (\partial \lambda_i/\partial \zeta)$. Pre-multiplying equation (6) by $\mathbf{v}_i^{\mathrm{T}}$ and using equation (5a) and the transpose of equation (4), yields [6]

$$\dot{\lambda}_i = \mathbf{v}_i^{\mathrm{T}} \dot{\mathbf{A}}\mathbf{u}_i. \tag{7}$$

Differentiating equation (7) again, and with the same notation,

$$\ddot{\lambda}_i = \dot{\mathbf{v}}_i^{\mathrm{T}} \dot{\mathbf{A}}\mathbf{u}_i + \mathbf{v}_i^{\mathrm{T}} \ddot{\mathbf{A}}\mathbf{u}_i + \mathbf{v}_i^{\mathrm{T}} \dot{\mathbf{A}}\dot{\mathbf{u}}_i. \tag{8}$$

Since $\mathbf{u}_j$, $j = 1, \ldots, n$, are assumed to be independent, then $\dot{\mathbf{u}}_i$ can be expressed in terms of $\mathbf{u}_j$, $j = 1, \ldots, n$, as

$$\dot{\mathbf{u}}_i = \sum_j \alpha_{ij} \mathbf{u}_j, \tag{9}$$

where the coefficients α_{ij} can be obtained by substituting expression (9) in equation (6) and using equations (3) and (5):

$$\alpha_{ij} = (\mathbf{v}_j^{\mathrm{T}} \dot{\mathbf{A}}\mathbf{u}_i)/(\lambda_i - \lambda_j), \qquad j \neq i. \tag{10}$$

Note that α_{ii} is not defined, as it is not needed in the subsequent analysis. By using a similar argument, $\dot{\mathbf{v}}_i$ can be expressed in terms of $\mathbf{v}_j$, $j = 1, \ldots, n$, as

$$\dot{\mathbf{v}}_i = \sum_j \gamma_{ij} \mathbf{v}_j, \tag{11}$$

where

$$\gamma_{ij} = -\alpha_{ji} = \mathbf{v}_i^{\mathrm{T}} \dot{\mathbf{A}} \mathbf{u}_j / (\lambda_i - \lambda_j), \qquad j \neq i. \tag{12}$$

Therefore, from equation (8)

$$\ddot{\lambda}_i = \mathbf{v}_i^{\mathrm{T}} \ddot{\mathbf{A}} \mathbf{u}_i + 2 \sum_{j \neq i} \alpha_{ij} \alpha_{ji} (\lambda_j - \lambda_i). \tag{13}$$

Equations (7) and (13) give the first and second order eigenvalue sensitivities as currently known in the literature with, sometimes, different methods of representation. The main computational effort is expended in calculating the coefficients α_{ij}. Using similar manipulations, as shown in the Appendix, yields the third-order sensitivities as

$$\dddot{\lambda}_i = \mathbf{v}_i^{\mathrm{T}} \dddot{\mathbf{A}} \mathbf{u}_i + 3 \sum_{j \neq i} [\alpha_{ij} (\mathbf{v}_i^{\mathrm{T}} \ddot{\mathbf{A}} \mathbf{u}_j) - \alpha_{ji} (\mathbf{v}_j^{\mathrm{T}} \ddot{\mathbf{A}} \mathbf{u}_i)] + 6 \sum_{j \neq i} \alpha_{ij} \alpha_{ji} (\dot{\lambda}_i - \dot{\lambda}_j)$$

$$+ 2 \sum_{j \neq i} \sum_{\substack{k \neq j \\ k \neq i}} (\alpha_{ij} \alpha_{jk} \alpha_{ki} - \alpha_{ji} \alpha_{ik} \alpha_{kj})(2\lambda_k - \lambda_i - \lambda_j). \tag{14}$$

3.2. RANK-1 SENSITIVITY FORMULAS

The application of the rank-1 matrix exchange formula (2) has a powerful impact on the eigenvalue sensitivity formulas derived in Section 3.1. First, consider the term $\alpha_{ij}\alpha_{ji}$ in the second order eigenvalue sensitivity expression of (13). Then, using equation (10), one has

$$\alpha_{ij} \alpha_{ji} = [(\mathbf{v}_j^{\mathrm{T}} \dot{\mathbf{A}} \mathbf{u}_i)(\mathbf{v}_i^{\mathrm{T}} \dot{\mathbf{A}} \mathbf{u}_j)] / [(\lambda_i - \lambda_j)(\lambda_j - \lambda_i)], \tag{15}$$

By letting $\mathbf{C} = \mathbf{D} = \dot{\mathbf{A}}$, $\mathbf{g} = \mathbf{u}_i$, $\mathbf{h} = \mathbf{v}_j$, $\mathbf{p} = \mathbf{u}_j$ and $\mathbf{q} = \mathbf{v}_i$, and using equation (2),

$$\alpha_{ij} \alpha_{ji} = [(\mathbf{v}_i^{\mathrm{T}} \dot{\mathbf{A}} \mathbf{u}_i)(\mathbf{v}_j^{\mathrm{T}} \dot{\mathbf{A}} \mathbf{u}_j)] / [(\lambda_i - \lambda_j)(\lambda_j - \lambda_i)] \tag{16}$$

and, from equation (7),

$$\alpha_{ij} \alpha_{ji} = \dot{\lambda}_i \dot{\lambda}_j / [(\lambda_i - \lambda_j)(\lambda_j - \lambda_i)]. \tag{17}$$

Hence, for a rank-1 $\dot{\mathbf{A}}$ matrix, equation (13) reduces to

$$\ddot{\lambda}_i = \mathbf{v}_i^{\mathrm{T}} \ddot{\mathbf{A}} \mathbf{u}_i + 2 \sum_{j \neq i} \dot{\lambda}_i \dot{\lambda}_j / (\lambda_i - \lambda_j): \tag{18}$$

that is, the second order sensitivities can be obtained directly by using the first order sensitivities already calculated.

The application of equation (2) can be extended easily to the third and higher order sensitivities. For example, for the third order sensitivity expression (14), first consider the term

$$\alpha_{ij}\alpha_{jk}\alpha_{ki} = [(\mathbf{v}_j^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_i)(\mathbf{v}_k^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_j)(\mathbf{v}_i^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_k)]/[(\lambda_i-\lambda_j)(\lambda_j-\lambda_k)(\lambda_k-\lambda_i)].$$

Applying equation (2) twice with $\mathbf{C} = \mathbf{D} = \dot{\mathbf{A}}$, one obtains

$$\alpha_{ij}\alpha_{jk}\alpha_{ki} = \dot{\lambda}_i\dot{\lambda}_j\dot{\lambda}_k/[(\lambda_i-\lambda_j)(\lambda_j-\lambda_k)(\lambda_k-\lambda_i)]. \tag{19}$$

Similarly, it can be shown that

$$\alpha_{ji}\alpha_{ik}\alpha_{kj} = -\alpha_{ij}\alpha_{jk}\alpha_{ki}. \tag{20}$$

Now, the term $\alpha_{ij}(\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j)$ of equation (14) can be reduced further if $\dot{\mathbf{A}}$ and $\ddot{\mathbf{A}}$ have the special relationship between the matrices $\mathbf{C}$ and $\mathbf{D}$, respectively, as in equations (1). In this case,

$$\alpha_{ij}(\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j) = (\mathbf{v}_j^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_i)(\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j)/(\lambda_i-\lambda_j) = (\mathbf{v}_i^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_i)(\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j)/(\lambda_i-\lambda_j),$$

and, from equations (7) and (18),

$$\alpha_{ij}(\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j) = \frac{\dot{\lambda}_i}{(\lambda_i-\lambda_j)}\left[\ddot{\lambda}_j - 2\sum_{k\neq j}\frac{\dot{\lambda}_j\dot{\lambda}_k}{(\lambda_j-\lambda_k)}\right]. \tag{21}$$

Similarly, it can be shown that

$$\alpha_{ji}(\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i) = \frac{\dot{\lambda}_j}{(\lambda_j-\lambda_i)}\left[\ddot{\lambda}_i - 2\sum_{k\neq i}\frac{\dot{\lambda}_i\dot{\lambda}_k}{(\lambda_i-\lambda_k)}\right]. \tag{22}$$

Upon using equation (17) and equations (19)–(22), the third order expression (14) reduces to

$$\dddot{\lambda}_i = \mathbf{v}_i^{\mathrm{T}}\dddot{\mathbf{A}}\mathbf{u}_i + 3\sum_{j\neq i}\left[\frac{(\dot{\lambda}_i\ddot{\lambda}_j + \dot{\lambda}_j\ddot{\lambda}_i)}{(\lambda_i-\lambda_j)}\right] - 2\sum_{j\neq i}\sum_{\substack{k\neq i\\ k\neq j}}\left[\frac{\dot{\lambda}_i\dot{\lambda}_j\dot{\lambda}_k(2\lambda_k-\lambda_i-\lambda_j)}{(\lambda_i-\lambda_j)(\lambda_j-\lambda_k)(\lambda_k-\lambda_i)}\right], \tag{23}$$

which, again, is recursive and depends solely on the previously computed first and second order sensitivities.

3.3. SPECIAL CASE

Consider the special case in which the sensitivity parameters represent elements a_{kl} of the system matrix itself [6]. In this case the first order sensitivity formula (7) reduces to

$$\dot{\lambda}_i = \partial\lambda_i/\partial a_{kl} = v_{ik}u_{il}. \tag{24}$$

Since $\dot{\mathbf{A}}$ in this case has zero elements in all but the kl location with $\dot{a}_{kl} = 1$, the rank-1 exchange formula (2) would apply, as it is always possible to find a matrix $\mathbf{C} = \dot{\mathbf{A}}$ by defining the elements of $\mathbf{z}$ as $z_k = 1$ and $z_j = 0$ for $j \neq k$ and, similarly, $x_l = 1$ and $x_j = 0$ for $j \neq l$. Therefore, since in this $\ddot{\mathbf{A}} = \mathbf{0}$ and $\alpha_{ij} = v_{jk}u_{il}/(\lambda_i-\lambda_j)$, the second order sensitivity formula (13) reduces to

$$\ddot{\lambda}_i = \partial^2\lambda_i/\partial a_{kl}^2 = 2\sum_{j\neq i}(u_{il}u_{jl}v_{ik}v_{jk})/(\lambda_i-\lambda_j), \tag{25}$$

which is the same result obtained by direct application of equation (18) and using equation (24). Similarly, the third order sensitivity can be obtained by direct application of equation (23), with $\ddot{\mathbf{A}} = \mathbf{0}$.

It is noted that the resulting sensitivity formulas for this special case reduce to very fast computational schemes involving scalar operations and using elements of the eigenvectors of the original system matrix and its transpose.

4. ASSESSMENT OF STORAGE AND CPU TIME

A simple comparison between the rank-1 sensitivity formulas (18) and (23) and the corresponding conventional formulas (13) and (14) would reveal that the rank-1 formulas require much less computer storage. This is mainly because they avoid the calculation and storage of the $[\alpha_{ij}]$ and $[\mathbf{v}_i^{\mathrm{T}} \ddot{\mathbf{A}} \mathbf{u}_j]$ coefficient matrices. On the other hand, with regard to CPU time requirements, the authors of this paper have analyzed several engineering systems of different sizes in order to assess the saving in computer time associated with the use of rank-1 matrix formulas for eigenvalue sensitivities as compared to conventional formulas. A total of 48 case studies involving six different engineering systems were analyzed. The applications included the design and operation of mechanical structures and control schemes as well as the dynamic stability of rotating electrical synchronous machines. In the studies performed, the order of the system matrix, which represents number of columns or rows, ranged from five to 500. Out of the 48 case studies analyzed, 41 cases exhibited rank-1 system matrix derivatives and were therefore considered for further comparison with conventional formulas. While the use of the rank-1 matrix formula for first order sensitivity calculation does not provide any savings, its use for second and third order sensitivities offers remarkable savings in computational time. Even when a very efficient computational scheme was employed at the expense of memory saving (for example, by computing various matrix–vector products once and storing them for multi-use in subsequent computations), the conventional formulas took about 2·5 times as much as rank-1 formulas (in terms of CPU time for both second and third order sensitivities) on a computer workstation for 5×5 system matrices. The ratio was 6·0 for 20×20 system matrices, 13·5 for 50×50 matrices, 26·5 for 100×100 matrices, 76·0 for 300×300 matrices and 126·0 for 500×500 matrices. Indeed, such savings would represent a significant improvement, especially when many design and operational parameters are to be considered in the sensitivity analysis. In the case of a 500×500 system matrix, the rank-1 formula took only 0·002 and 0·018 of the CPU time taken by conventional formulas when calculating second and third order sensitivities, respectively.

During the studies performed, it was noted that the use of eigenvalue sensitivities to estimate changes in system modes for large parameter changes would lead to sufficiently accurate results and, therefore, repeat eigenvalue calculations could be avoided. The size of the estimation error would depend on the size of the parameter change and the order of sensitivity used. In one dynamic stability application with a +30% change in a system gain parameter, the percentage error in estimating the change in the dominant eigenvalue dropped from 16·9% to 5·3% as the sensitivity order increased from first to second. The error was reduced further to 2·4% when third order sensitivities were employed.

6. CONCLUSIONS

The general rank-1 matrix formula used in this paper allows for a certain rearrangement of terms in multi-product forms involving vectors and matrices. For exact (non-approximate) solutions to be obtained, the application of the formula requires that the matrices in the product form possess certain rank-1 structures. Nevertheless, such special structures of matrices are believed to exist in many engineering applications and the use of the formula would be very useful in such applications. In applying the formula to the eigenvalue sensitivity problem, as was demonstrated in the paper, the eigenvalue sensitivity expressions can be reduced to compact, fast and elegant forms which are recursive in nature and can be easily coded in computer programs. The special case in which the sensitivity parameter of interest is simply an element of the original matrix was presented in the paper. In this case, the resulting formulas for first and higher-order eigenvalue sensitivities reduce to very fast, schemes involving scalar operations using elements of the eigenvectors of the original system matrix and its transpose.

Analysis of several engineering systems of different sizes has shown that the rank-1 sensitivity formulas provide savings in the order of 60% for 5×5 system matrices in the CPU time required to calculate second and third order sensitivities. Such savings are increased to 83% for 20×20 matrices, to 93% for 50×50 matrices, to 96% for 100×100 matrices, to 98·7% for 300×300 matrices and to 99·2% for 500×500 matrices. These savings are realized for each sensitivity parameter of interest, leading to a remarkable reduction in the overall computational time. For many engineering systems analyzed by the authors, very good estimates of eigenvalue changes could be obtained for changes in system parameters of up to $\pm 25\%$ without the need to repeat the entire eigenvalue calculations. It is also noted that the use of third order sensitivities would, in some cases, constitute a major improvement in accuracy.

REFERENCES

1. M. A. Laughton 1964 *Journal of Electronic Control* **17**, 557–591. Sensitivity in dynamical systems analysis.
2. J. E. Van Ness, J. M. Boyle and F. P. Imad 1965 *IEEE Transactions on Automatic Control* **10**, 308–318. Sensitivity of large multi-loop systems.
3. T. R. Crossley and B. Porter 1969 *International Journal of Control* **10**, 163–170. Eigenvalue and eigenvector sensitivities in linear systems theory.
4. E. J. Haugh and B. Rousselt 1980 *Journal of Structural Mechanics* **8**, 161–186. Design sensitivity analysis in structural mechanics.
5. P. Vanhonacker 1980 *American Institute of Aeronautics and Astronautics Journal* **18**, 970–974. Differential and difference sensitivities of natural frequencies and mode shapes of mechanical structures.
6. P. C. Brooks and R. S. Sharp 1987 *Journal of Sound and Vibration* **114**, 13–18. A computational procedure based on eigenvalue sensitivity theory applicable to linear system design.
7. R. Z. Zimoch 1991 *Journal of Sound and Vibration* **145**, 151–157. Fast computation of sensitivities and eigenvalues for systems with non-symmetric structural matrices.
8. Z. Elrazaz and A. Al-Ohaly 1993 *IEEE Transactions on Power Systems* **8**, 1545–1549. Criterion for inductive compensation location to enhance system steady state stability.
9. M. A. El-Kady and A. Al-Ohaly 1995 *Technical Report*, *King Saud University*. A rank-1 matrix exchange formula with applications to eigenvalue sensitivity computations.

APPENDIX: DERIVATION OF THIRD ORDER SENSITIVITIES

In order to derive third order eigenvalue sensitivities, the second order sensitivities of equation (13) are differentiated, yielding

$$\dddot{\lambda}_i = 2 \sum_{j \neq i} [\dot{\alpha}_{ij}\alpha_{ji}(\lambda_j - \lambda_i) + \alpha_{ij}\dot{\alpha}_{ji}(\lambda_j - \lambda_i) + \alpha_{ij}\alpha_{ji}(\dot{\lambda}_j - \dot{\lambda}_i)$$

$$+ \sum_{j \neq i} [\gamma_{ij}\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i + \alpha_{ij}\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j] + \mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i. \qquad \text{(A1)}$$

Now $\alpha_{ij}(\lambda_i - \lambda_j) = \mathbf{v}_j^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_i$. Hence,

$$\begin{aligned}
\dot{\alpha}_{ij}(\lambda_i - \lambda_j) &= -\alpha_{ij}(\dot{\lambda}_i - \dot{\lambda}_j) + \sum_k [\gamma_{jk}\mathbf{v}_k^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_i + \alpha_{ik}\mathbf{v}_j^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_k] + \mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i \\
&= -(\dot{\lambda}_i - \dot{\lambda}_j)\alpha_{ij} + \sum_k [-\alpha_{kj}\mathbf{v}_k^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_i + \alpha_{ik}\mathbf{v}_j^{\mathrm{T}}\dot{\mathbf{A}}\mathbf{u}_k] + \mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i \\
&= -(\dot{\lambda}_i - \dot{\lambda}_j)\alpha_{ij} + \mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i + \sum_{\substack{k \neq i \\ k \neq j}} [-\alpha_{kj}\alpha_{ik}(\lambda_i - \lambda_k) + \alpha_{ik}\alpha_{kj}(\lambda_k - \lambda_j)] \\
&\quad + [-\alpha_{ij}\dot{\lambda}_i + \alpha_{ii}\,\alpha_{ij}(\lambda_i - \lambda_j)] + [-\alpha_{jj}\alpha_{ij}(\lambda_i - \lambda_j) + \alpha_{ij}\dot{\lambda}_{ij}] \\
&= -2(\dot{\lambda}_i - \dot{\lambda}_j)\alpha_{ij} + \mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i + \sum_{\substack{k \neq i \\ k \neq j}} [-\alpha_{kj}\alpha_{ik}(\lambda_i - \lambda_k) + \alpha_{ik}\alpha_{kj}(\lambda_k - \lambda_j)] \\
&\quad + \alpha_{ij}(\lambda_i - \lambda_j)[\alpha_{ii} - \alpha_{jj}].
\end{aligned}$$

Similarly,

$$\begin{aligned}
(\lambda_j - \lambda_i)\dot{\alpha}_{ji} &= -2(\dot{\lambda}_j - \dot{\lambda}_i)\alpha_{ji} + \mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j \\
&\quad + \sum_{\substack{k \neq j \\ k \neq i}} [-\alpha_{ki}\alpha_{jk}(\lambda_j - \lambda_k) + \alpha_{jk}\,\alpha_{ki}(\lambda_k - \lambda_i)] + \alpha_{ji}(\lambda_j - \lambda_i)[\alpha_{jj} - \alpha_{ii}].
\end{aligned}$$

Hence, equation (A1) can be written as

$$\begin{aligned}
\ddot{\lambda}_i &= \sum_j \gamma_{ij}\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i + \sum_j \alpha_{ij}\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j + \mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i \\
&\quad + 2\sum_{j \neq i} -\alpha_{ji}\dot{\alpha}_{ij}(\lambda_i - \lambda_j) + 2\sum_{j \neq i} \alpha_{ij}\dot{\alpha}_{ji}(\lambda_j - \lambda_i) + 2\sum_{j \neq i} \alpha_{ij}\alpha_{ji}(\dot{\lambda}_j - \dot{\lambda}_i) \\
&= \sum_j [-\alpha_{ji}\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i + \alpha_{ij}\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j] + \mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i - 2\sum_{j \neq i} \alpha_{ji}\{-2(\dot{\lambda}_i - \dot{\lambda}_j)\alpha_{ij} + \mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i \\
&\quad + \sum_{\substack{k \neq i \\ k \neq j}} [-\alpha_{kj}\alpha_{ik}(\lambda_i - \lambda_k) + \alpha_{ik}\alpha_{kj}(\lambda_k - \lambda_j)] + \alpha_{ij}(\lambda_i - \lambda_j)(\alpha_{ii} - \alpha_{jj})\} \\
&\quad + 2\sum_{j \neq i} \alpha_{ij}\{-2(\dot{\lambda}_j - \dot{\lambda}_i)\alpha_{ji} + \mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j + \sum_{\substack{k \neq j \\ k \neq i}} [-\alpha_{ki}\alpha_{jk}(\lambda_j - \lambda_k) + \alpha_{jk}\alpha_{ki}(\lambda_k - \lambda_i)] \\
&\quad + \alpha_{ji}(\lambda_j - \lambda_i)(\alpha_{jj} - \alpha_{ii})\} + 2\sum_{j \neq i} \alpha_{ij}\alpha_{ji}(\dot{\lambda}_j - \dot{\lambda}_i).
\end{aligned}$$

Rearranging terms then yields

$$\ddot{\lambda}_i = \mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i + \sum_{j\neq i}[\alpha_{ij}\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j - \alpha_{ji}\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i] + [\alpha_{ii}\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i - \alpha_{ii}\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i]$$

$$+ 4\sum_{j\neq i}\alpha_{ij}\alpha_{ji}(\dot{\lambda}_i - \dot{\lambda}_j) - 2\sum_{j\neq i}\alpha_{ji}\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i$$

$$- 2\sum_{j\neq i}\sum_{\substack{k\neq i\\ k\neq j}}[-\alpha_{ji}\alpha_{kj}\alpha_{ik}(\lambda_i - \lambda_k) + \alpha_{ji}\alpha_{ik}\alpha_{kj}(\lambda_k - \lambda_j)]$$

$$- 2\sum_{j\neq i}[\alpha_{ji}\alpha_{ij}(\lambda_i - \lambda_j)(\alpha_{ii} - \alpha_{jj})] + 4\sum_{j\neq i}\alpha_{ij}\alpha_{ji}(\dot{\lambda}_i - \dot{\lambda}_j) + 2\sum_{j\neq i}\alpha_{ij}\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j$$

$$+ \sum_{j\neq i}\sum_{\substack{j\neq j\\ k\neq i}}[-\alpha_{ij}\alpha_{ki}\alpha_{jk}(\lambda_j - \lambda_k) + 2\alpha_{ij}\alpha_{jk}\alpha_{ki}(\lambda_k - \lambda_i)]$$

$$+ 2\sum_{j\neq i}\alpha_{ij}\alpha_{ji}(\lambda_j - \lambda_i)(\alpha_{jj} - \alpha_{ii}) + 2\sum_{j\neq i}\alpha_{ij}\alpha_{ji}(\dot{\lambda}_j - \dot{\lambda}_i)$$

Hence,

$$\ddot{\lambda}_i = \mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i + 3\sum_{j\neq i}[\alpha_{ij}(\mathbf{v}_i^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_j) - \alpha_{ji}(\mathbf{v}_j^{\mathrm{T}}\ddot{\mathbf{A}}\mathbf{u}_i)] + 6\sum_{j\neq i}\alpha_{ij}\alpha_{ji}(\dot{\lambda}_i - \dot{\lambda}_j)$$

$$+ 2\sum_{j\neq i}\sum_{\substack{k\neq j\\ k\neq i}}(\alpha_{ij}\alpha_{jk}\alpha_{ki} - \alpha_{ji}\alpha_{ik}\alpha_{kj})(2\lambda_k - \lambda_i - \lambda_j).$$