JOURNAL OF AIRCRAFT
Vol. 43, No. 4, July–August 2006

On the Formulation of Minimum-State Approximation as a Nonlinear Optimization Problem

E. Nissim*
University of California, Los Angeles, California 90095-1597

The solution of the minimum-state (MS) approximation of the unsteady aerodynamic forces is brought in this work into the form of a nonlinear optimization problem. This new formulation takes as design variables all of the aerodynamic lag terms (known also as aerodynamic roots) as well as the two matrices that directly operate on these lag terms. This new formulation enables the explicit determination of the remaining matrices that form the MS approximation, and it does not require enforcing any equality constraints. Furthermore, it also permits the derivation of simple analytical expressions for the gradients of the least-square (LS)-type objective function. This combination of explicit expressions for both the gradients and some of the unknown matrices leads to a dramatic reduction in computational labor. It is also shown that by appropriately scaling the tabulated aerodynamic matrix a significantly accelerated rate of convergence is obtained during the process of optimization, whereas a general weighting scheme might considerably slow down this convergence. It is also shown that the preceding scaling of the tabulated aerodynamic matrix can also significantly reduce the computational labor required by current methods of solution that are based on iterative LS analysis. The new formulation presented in this work leads to better results (i.e., lower values for the objective function) than those obtained using current iterative LS-based methods that use preassumed values for the aerodynamic lag terms. At the same time, the computer CPU time required by these two methods of solution is of the same order, with only slightly higher CPU values needed for the new formulation. On the other hand, current methods based on iterative LS analysis that attempt to optimize the aerodynamic roots rather than use preassumed values need extensive added computational labor to the extent that makes them practically unattractive.

Nomenclature

$\boldsymbol{A}$	=	tabulated matrix of generalized aerodynamic coefficients, of order $n \times m$
$\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2$	=	coefficient matrices of rational approximation, of order $n \times m$ [Eq. (1)]
$\bar{\boldsymbol{A}}$	=	defined in Eq. (4)
$\bar{\bar{\boldsymbol{A}}}$	=	defined in Eq. (6)
$a_{0,ij}, a_{1,ij}, a_{2,ij}$	=	i, j element of matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2$, respectively
a_{ij}	=	i, j element of matrix $\boldsymbol{A}$
$\bar{a}_{ij}$	=	i, j element of matrix $\bar{\boldsymbol{A}}$
$\bar{\bar{a}}_{ij}$	=	i, j element of matrix $\bar{\bar{\boldsymbol{A}}}$
$colmax(j)$	=	absolute maximum value of the jth column of $\boldsymbol{A}$
$\boldsymbol{D}$	=	coefficient matrix in rational approximation, of order $n \times nl$ [Eq. (1)]
d_{ij}	=	i, j element of matrix $\boldsymbol{D}$
$\bar{d}_{ij}$	=	defined in Eq. (18)
$\boldsymbol{E}$	=	coefficient matrix in rational approximation, of order $nl \times m$ [Eq. (1)]
e_{ij}	=	i, j element of matrix $\boldsymbol{E}$
$\bar{e}_{ij}$	=	defined in Eq. (18)
F	=	objective function
k	=	reduced frequency
k_f	=	value of k at which scaling is applied to matrix $\boldsymbol{A}$
k_r	=	rth reduced frequency of the tabulated $\boldsymbol{A}$
m	=	number of columns in aerodynamic matrices
n	=	number of rows in aerodynamic matrices
nit	=	number of minimum-state iterations
$nitmax$	=	number of iterations during optimization, as a multiple of the number of design variables
nk	=	number of reduced frequencies in the tabulated values of $\boldsymbol{A}$
nl	=	number of assumed aerodynamic roots
$rowmax(i)$	=	maximum absolute value of the ith row of $\boldsymbol{A}$, after normalizing its columns
u_i	=	defined in Eq. (13)
v_j	=	defined in Eq. (13)
w_{ij}	=	weighting given to the ijth element of $\boldsymbol{A}$, defined in Eq. (12)
w_{ijk_r}	=	weighting given to the ijth element of $\boldsymbol{A}$ at the rth tabulated reduced frequency
γ	=	matrix of aerodynamic lag terms, defined in Eq. (2)
γ_l	=	lth aerodynamic root

Superscript

s	=	relates to scaled aerodynamic matrices

Introduction

AEROELASTIC design and analysis often require casting the equations of motion into a linear, time-invariant state-space form. This requirement has led to several rational functions approximations (RFA) of the aerodynamic matrix in the Laplace domain. The most widely used techniques are those based on the least-square (LS) method, which include Roger's matrix Pade[1,2] method, the modified matrix Pade method,[3] and the minimum-state (MS) method.[4–6] The resulting state-space equations include augmented states that represent the aerodynamic lags or the aerodynamic roots of the RFA. The number of aerodynamic augmented states resulting from the Roger's approximation is equal to the number of modes (or rows of the aerodynamic matrix) multiplied by the number of assumed aerodynamic roots. The number of aerodynamic augmented states resulting from the modified matrix Pade method is equal to the sum, over all of the columns of the aerodynamic matrix, of the aerodynamic roots assumed in each of the columns. The number of

Received 12 April 2005; accepted for publication 29 September 2005.

*Professor Emeritus, Faculty of Engineering, Department of Mechanical and Aerospace Engineering; currently at Technion—Israel Institute of Technology, 3200 Haifa, Israel.

augmented aerodynamic states resulting from the MS approximation is equal to the number of assumed aerodynamic roots and is independent of the size of the aerodynamic matrix. However, unlike the other methods, the determination of the matrices involved in the MS approximation is relatively costlier because it involves iterative double LS analysis. Comparative assessment of the different methods[7] indicates that for a given number of augmented aerodynamic states the best approximation of the aerodynamics forces, in terms of the LS overall errors, is obtained when using the MS approximation. The computational cost involved in the determination of any of the just-mentioned representations, although important in itself, represents only a part of a bigger picture because some of these costs can ultimately be offset by the resulting smaller-order equations, which, in turn, need less computer time to solve. Therefore, the present work will limit itself to the MS approximation with the object of improving the approximation for any specific chosen number augmented states. In this respect, the following two main directions that had already been investigated in the past will be tackled:

1) Introduction of flexibilities in the equality constraints: To reduce computational labor, equality constraints were introduced[4] in the early developments of the MS method. This implies that the MS approximation is forced to yield identical values to those of the tabulated aerodynamic matrix at some chosen values of the reduced frequency k. This leads to a considerable reduction in the order of the iterative LS problem and thus to a significant reduction in computational labor.[6,7] However, this is achieved at the expense of a significant increase in the LS errors of the resulting approximation. Karpel and Strul[6] took advantage of the special form of the resulting LS equations, and thus managed to reduce the computational labor required by the unconstrained approximation by around 70%. Even so, this streamlined unconstrained analysis still required about three times more computational time than the constrained approximation. It has further been shown[6] that the constraint imposed at $k = 0$ had the *least* effect on the overall approximation error, whereas the other equality constraints showed major effects.

2) Optimization of the aerodynamic roots: Most current methods to determine the MS approximation use preassumed aerodynamic roots. Dunn[2] optimized the values of the aerodynamic roots in the Roger's method and showed that this resulted in a significant reduction of the approximation errors. Karpel[4] combined the iterative LS method of solution with an optimization procedure based on Davidon's method. However, to avoid extensive computational labor he limited his work to a fully constrained MS approximation, which imposed three equality constraints. In a later work Tiffany and Adams[7] developed an extended MS approximation, which they called EMS. This extended MS approximation not only introduced flexibilities in the number of equality constraints (from a fully constrained approximation to a completely unconstrained one), but also permitted the optimization of the aerodynamic roots in a fashion similar to the one mentioned earlier.[4] Their experience was "that applying this optimization to the MS method requires considerably more computation time than applying it to the other methods since it adds another iteration process to a method that already requires a two-step iteration process." They reported that this increase in computation time varied between 2500 to 3000 times (!!!) the baseline cost (i.e., of a fully constrained with preassumed aerodynamic roots). They therefore concluded that it would be much more economical to reduce the MS approximation errors by increasing the number of the preassumed aerodynamic roots of the constrained MS approximation (thus increasing the order of the resulting equations), rather than to optimize the values of these roots and introduce flexibilities into the constraints.

Beyond the points just mentioned, extensive use has been made of weighting during the MS iterative LS process. This point will be discussed in a later section of this work.

It should be made clear at this stage that the main purpose of this work is to formulate a computationally *efficient* method for the unconstrained MS approximation, based *entirely* on nonlinear optimization, which will permit to include among its many design variables also the values of the aerodynamic roots.

Mathematical Formulation

Let the MS approximation of $\boldsymbol{A}$, as a function of k, be given by

$$\boldsymbol{A} \approx \boldsymbol{A}_0 + ik\boldsymbol{A}_1 - k^2\boldsymbol{A}_2 + \boldsymbol{D}\boldsymbol{\gamma}\boldsymbol{E} \tag{1}$$

where $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{D}$, and $\boldsymbol{E}$ are all real matrices, and $\boldsymbol{\gamma}$ is a diagonal matrix defined by

$$\boldsymbol{\gamma} = \begin{bmatrix} ik/(ik+\gamma_1) & & & \\ & ik/(ik+\gamma_2) & & \\ & & \ddots & \\ & & & ik/(ik+\gamma_{nl}) \end{bmatrix} \tag{2}$$

where $i = \sqrt{-1}$. Matrix $\boldsymbol{\gamma}$ will be referred to as the aerodynamic lags matrix (of order $nl \times nl$) and the γ_l terms as the aerodynamic lags or the aerodynamic roots. It will further be assumed that the values of matrix $\boldsymbol{A}$ are known over a range of nk reduced frequencies, from $k = 0$ to k_n. Because Eq. (1) is only an approximate representation of $\boldsymbol{A}$ as a function of k, one needs to determine the values of matrices $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{D}, \boldsymbol{E}$, and $\boldsymbol{\gamma}$ so as to minimize the residuals of Eq. (1) over the preceding range of k values. The conventional method used for the determination of the preceding matrices is based on bringing Eq. (1) into the form of a least-squares problem by 1) assuming fixed values for the nl lag terms γ_l in $\boldsymbol{\gamma}$; 2) assuming initial values for $\boldsymbol{D}$ and solving for $\boldsymbol{E}$, and then using the solution for $\boldsymbol{E}$ to determine $\boldsymbol{D}$ in an iterative $\boldsymbol{D} \to \boldsymbol{E} \to \boldsymbol{D}$ manner; 3) introducing some equality constraints to facilitate the determination of matrices $\boldsymbol{A}_0, \boldsymbol{A}_1$, and $\boldsymbol{A}_2$, so as to reduce the iterative computational labor; and 4) because weighting is invariably used in solving the linear least-squares problem, matrix $\boldsymbol{E}$ is determined a column at a time, followed by a similar determination of $\boldsymbol{D}$, a row at a time.

At this stage, we can restate the purpose of this work, that is, to formulate a computationally efficient solution for the MS matrices that is based entirely on nonlinear optimization, without having to preassume the values for the γ_l terms, and without having to introduce the equality constraints mentioned in point 3. Based on previous investigations, such a formulation should yield the lowest residual errors for any given number of lag terms, and it might justify itself only if in so doing the added computational labor reported earlier[7] is drastically reduced.

Some Preliminary Considerations

The residuals of Eq. (1), over the range of nk values at which the exact values of $\boldsymbol{A}$ are known, give the best indication regarding the quality of the approximation. The smaller the absolute values of the residuals, the better the approximation is. Therefore, the formulation of the objective function should involve the sum of the squares of all the residuals of Eq. (1) at the nk values of k at which $\boldsymbol{A}$ is known. Furthermore, because it is desired to use the MS representation to solve also static aeroelastic problems, in addition the to rigid-body contributions to the dynamics of the flying vehicle, it is very important to have an exact fit of Eq. (1) at $k = 0$. This implies that the single constraint whereby

$$\boldsymbol{A}_0 = \boldsymbol{A}_{k=0} \tag{3}$$

is *essential* to the approximation, disregarding its possible effects on the resulting computational labor. In general, the number of the unknown coefficients in Eq. (1) can be very large, depending clearly on the order of $\boldsymbol{A}$. Therefore, the number of objective function evaluations during the process of optimization can become unreasonably large, especially if we let the gradients of the objective function (OF) F be determined by finite differencing. To avoid such a situation, the desired formulation should permit a relatively simple analytical determination of the gradients of the F. This requirement leads to a formulation of the optimization problem whereby *all* of the elements of $\boldsymbol{D}$, $\boldsymbol{E}$, and $\boldsymbol{\gamma}$ are taken as design variables, whereas all of the elements of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are solved explicitly using linear-least-squares (LLS) analysis. If needed, an explicit expression for $\boldsymbol{A}_0$ can readily be derived with negligible additional cost. It will be shown that such a formulation avoids any matrix inversions during the LLS solution process involved in the determination of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. It follows that the number of design variables involved in this formulation is $nl \times (n+m+1)$. The analytical computation of the second derivatives of F was ruled out (although very simple to derive analytically under the just-stated assumptions) because such a determination will eventually require the inversion of the Hessian matrix, which, as already stated, can be of very large order and therefore will require intensive computational labor. Instead, optimization algorithms that progressively estimate the inverse of the Hessian matrix [such as the Davidon–Fletcher–Powell (DFP) algorithm] will be used in this work. All of the design variables in this problem are unconstrained during optimization except for the γ_i lag terms that are constrained by the optimization routine to assume positive values.

On the Use of Weighting During Optimization

The current MS method uses weighted least squares for the determination of $\boldsymbol{D}$ and $\boldsymbol{E}$. Weighting is generally introduced for two reasons:

1) The first is to avoid the dominance of some large aerodynamic terms during the iterative LS fit of Eq. (1) and to counter the effects of normalization during the process of obtaining the generalized aerodynamic matrix $\boldsymbol{A}$. In practice, weighting has been applied in such a way as to effectively turn each aerodynamic term equally important during the iterative LS analysis, with the exception of those terms with absolute values smaller than 1. This is often referred to as normalizing weights. Three main comments can be made regarding the normalizing weights. First, it can be readily argued against the preceding choice of the absolute values being smaller than 1 because problems can be easily scaled in such a way so that *all* of the aerodynamic terms become smaller than 1. Does this imply that no weighting is needed? Second, it can be argued against turning all aerodynamic terms equally important during the iterative LS analysis because some aerodynamic terms are indeed relatively small, and they should rightfully be allowed to carry less weight than other aerodynamic terms with large numerical values. Third, to counter the effects of normalization during the process of obtaining the generalized aerodynamic matrix $\boldsymbol{A}$ one needs to be limited only to operations involving whole columns and whole rows of matrix $\boldsymbol{A}$, rather than the unrelated operations on the different elements of $\boldsymbol{A}$.

2) The second is to purposefully sacrifice the accuracy of the fit so as to improve the accuracy of flutter prediction (such as the physical weighting). This "tuning" of the aerodynamic fit is obviously configuration sensitive and requires the reevaluation of the aerodynamic MS representation every time that a flight configuration is changed. In this work, we shall avoid such a tuning with the objective of achieving maximum accuracy of the approximation, for all flight configurations, for any assumed number nl.

Despite the preceding remarks, it is clear that some type of weighting is often needed in order to reduce the dominance of the large aerodynamic elements that might be present in matrix $\boldsymbol{A}$. This is true irrespective of whether LS analysis or optimization procedures are to be employed to determine the coefficients in the MS representation. As already stated earlier, this problem will be addressed to in a later section of this work, where it will be shown that weighting might in general have an adverse effect on the rate of convergence when employing nonlinear optimization methods.

Formulation of the Objective Function

Define

$$\bar{\boldsymbol{A}} = \boldsymbol{A} - \boldsymbol{A}_0 \tag{4}$$

And let $\bar{a}_{ij}, a_{1,ij}, a_{2,ij}, d_{ij}, e_{ij}$ represent the elements of $\bar{\boldsymbol{A}}, \boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{D}, \boldsymbol{E}$ matrices, respectively. The OF F will be defined as the sum of the squares of the residuals of Eq. (1), that is,

$$\begin{aligned} F = {} & \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk} w_{ijk_r}^2 \Bigg[\mathrm{Re}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r^2}{\left(k_r^2+\gamma_l^2\right)} e_{lj} + k_r^2 a_{2,ij}\Bigg]^2 \\ & + \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk} w_{ijk_r}^2 \Bigg[\mathrm{Im}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{\gamma_l k_r}{\left(k_r^2+\gamma_l^2\right)} e_{lj} - k_r a_{1,ij}\Bigg]^2 \end{aligned} \tag{5}$$

Because at any moment during optimization the elements of matrices $\boldsymbol{D}$, $\boldsymbol{E}$, and $\boldsymbol{\gamma}$ are all known, one can determine $a_{1,ij}$ and $a_{2,ij}$ by using LLS applied to the known matrix $\bar{\bar{\boldsymbol{A}}}$, where

$$\bar{\bar{\boldsymbol{A}}} = \bar{\boldsymbol{A}} - \boldsymbol{D}\boldsymbol{\gamma}\boldsymbol{E} \tag{6}$$

It can be readily shown that the following expressions are the result of the LS analysis:

$$a_{1,ij} = \sum_{r=2}^{nk} w_{ijk_r}^2 \left[k_r\ \mathrm{Im}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r^2\gamma_l}{k_r^2+\gamma_l^2} e_{lj}\right] \Bigg/ \sum_{r=2}^{nk} k_r^2 w_{ijk_r}^2 \tag{7}$$

and

$$a_{2,ij} = \sum_{r=2}^{nk} w_{ijk_r}^2 \left[-k_r^2\ \mathrm{Re}(\bar{a}_{ij})_{k=k_r} + \sum_{l=1}^{nl} d_{il}\frac{k_r^4}{k_r^2+\gamma_l^2} e_{lj}\right] \Bigg/ \sum_{r=2}^{nk} k_r^4 w_{ijk_r}^2 \tag{8}$$

Differentiation of Eq. (5) while using Eqs. (7) and (8) leads to the following expressions for the derivatives of F with respect to the design variables:

$$\begin{aligned} \frac{\partial F}{\partial d_{il}} = {} & 2\sum_{j=1}^{m}\sum_{r=2}^{nk} w_{ijk_r}^2 \left[\mathrm{Re}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r^2}{k_r^2+\gamma_l^2} e_{lj} + k_r^2 a_{2,ij}\right] \\ & \times \left[-\frac{k_r^2}{k_r^2+\gamma_l^2} + \frac{k_r^2}{\sum_{r=2}^{nk} k_r^4 w_{ijk_r}^2}\left(\sum_{r=2}^{nk} w_{ijk_r}^2 \frac{k_r^4}{k_r^2+\gamma_l^2}\right)\right] e_{lj} \\ & + 2\sum_{j=1}^{m}\sum_{r=2}^{nk} w_{ijk_r}^2 \left[\mathrm{Im}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r\gamma_l}{k_r^2+\gamma_l^2} e_{lj} - k_r a_{1,ij}\right] \\ & \times \left[-\frac{k_r\gamma_l}{k_r^2+\gamma_l^2} + \frac{k_r}{\sum_{r=2}^{nk} k_r^2 w_{ijk_r}^2}\left(\sum_{r=2}^{nk} w_{ijk_r}^2 \frac{k_r^2\gamma_l}{k_r^2+\gamma_l^2}\right)\right] e_{lj} \end{aligned} \tag{9}$$

$$\begin{aligned} \frac{\partial F}{\partial e_{lj}} = {} & 2\sum_{i=1}^{n}\sum_{r=2}^{nk} w_{ijk_r}^2 \left[\mathrm{Re}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r^2}{k_r^2+\gamma_l^2} e_{lj} + k_r^2 a_{2,ij}\right] \\ & \times \left[-\frac{k_r^2}{k_r^2+\gamma_l^2} + \frac{k_r^2}{\sum_{r=2}^{nk} k_r^4 w_{ijk_r}^2}\left(\sum_{r=2}^{nk} w_{ijk_r}^2 \frac{k_r^4}{k_r^2+\gamma_l^2}\right)\right] d_{il} \\ & + 2\sum_{i=1}^{n}\sum_{r=2}^{nk} w_{ijk_r}^2 \left[\mathrm{Im}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r\gamma_l}{k_r^2+\gamma_l^2} e_{lj} - k_r a_{1,ij}\right] \\ & \times \left[-\frac{k_r\gamma_l}{k_r^2+\gamma_l^2} + \frac{k_r}{\sum_{r=2}^{nk} k_r^2 w_{ijk_r}^2}\left(\sum_{r=2}^{nk} w_{ijk_r}^2 \frac{k_r^2\gamma_l}{k_r^2+\gamma_l^2}\right)\right] d_{il} \end{aligned} \tag{10}$$

and

$$\frac{\partial F}{\partial \gamma_l} = 2\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk} w_{ijk_r}^2 \Bigg[\mathrm{Re}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r^2}{k_r^2+\gamma_l^2}e_{lj} + k_r^2 a_{2,ij}\Bigg]\Bigg\{\frac{2\gamma_l k_r^2}{\left(k_r^2+\gamma_l^2\right)^2} - \frac{k_r^2}{\sum_{r=2}^{nk} k_r^4 w_{ijk_r}^2}\Bigg[\sum_{r=2}^{nk} w_{ijk_r}^2 \frac{2\gamma_l k_r^4}{\left(k_r^2+\gamma_l^2\right)^2}\Bigg]\Bigg\} d_{il}e_{lj}$$
$$+ 2\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk} w_{ijk_r}^2 \Bigg[\mathrm{Im}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} d_{il}\frac{k_r\gamma_l}{k_r^2+\gamma_l^2}e_{lj} - k_r a_{1,ij}\Bigg] \times\Bigg\{\frac{2k_r\gamma_l^2}{\left(k_r^2+\gamma_l^2\right)^2} - \frac{k_r}{k_r^2+\gamma_l^2} - \frac{k_r}{\sum_{r=2}^{nk} k_r^2 w_{ijk_r}^2} \times \sum_{r=2}^{nk} w_{ijk_r}^2\Bigg[\frac{2k_r^2\gamma_l^2}{\left(k_r^2+\gamma_l^2\right)^2} - \frac{k_r^2}{k_r^2+\gamma_l^2}\Bigg]\Bigg\} d_{il}e_{lj} \tag{11}$$

Initial applications of Eqs. (5–11) to a numerical example showed extremely slow rates of convergence. The reason for this slow convergence is inherently connected to the weighting given to the different elements of F. To illustrate this point, consider that we have a very large aerodynamic term that requires a very small weighting, which is of order ε. The contribution of this term to the any of the preceding gradients will be even smaller, of order ε^2. On the other hand, a much smaller aerodynamic term with a unit weighting, for example, might turn to have a dominant contribution to the gradients, thus dominating the changes in the values of the design variables during the optimization process and leading to a very slow convergence. To overcome this problem, weighting based on the scaling of the aerodynamic matrix will next be considered.

Weighting Based on the Scaling of *A*

By the scaling of matrix $\boldsymbol{A}$, it is implied that we scan each column of $\boldsymbol{A}$ to determine its maximum absolute value [*colmax*(j), for $j = 1, \ldots, m$]. We then divide each column j of matrix $\boldsymbol{A}$ by its respective *colmax*(j) such that the maximum value of each column is reduced to 1. We then proceed to apply an identical procedure on the rows of matrix $\boldsymbol{A}$ in order to determine the maximum absolute value of each row [denoted by *rowmax*(i), with $i = 1, \ldots, n$]. We then divide each row by its respective *rowmax*(i) value. The resulting $\boldsymbol{A}$ matrix will thus have a maximum value of each of its rows and the maximum value of each of its columns equal to 1. Because $\boldsymbol{A}$ is a function of k, we need to choose a k value, say k_f, at which the preceding procedure is performed (or alternatively, scan, over all of the reduced frequencies for which $\boldsymbol{A}$ is tabulated, the columns and rows of $\boldsymbol{A}$ in the preceding indicated fashion). This is equivalent to the following weighting (which is independent of k) applied to matrix $\boldsymbol{A}$:

$$w_{ijk_r} = w_{ij} = [1/colmax(j)] \times [1/rowmax(i)] = v_j \times u_i \tag{12}$$

where clearly

$$u_i = 1/rowmax(i), \qquad v_j = 1/colmax(j) \tag{13}$$

As already stated earlier, these row and column operations are within the legitimate operations allowed if one wishes to counteract the nonunique effects resulting from the computation of the generalized aerodynamic matrix $\boldsymbol{A}$. Substitution of Eqs. (12) and (13), which are independent of k, into Eq. (5) yields

$$F = \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk}\Bigg[\mathrm{Re}(w_{ij}\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} u_i d_{il}\frac{k_r^2}{\left(k_r^2+\gamma_l^2\right)}v_j e_{lj} + k_r^2 w_{ij}a_{2,ij}\Bigg)^2\Bigg]^2 + \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk}\Bigg[\mathrm{Im}(w_{ij}\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl} u_i d_{il}\frac{\gamma_l k_r}{\left(k_r^2+\gamma_l^2\right)}v_j e_{lj} - k_r w_{ij}a_{1,ij}\Bigg)^2\Bigg]^2 \tag{14}$$

where

$$w_{ij}a_{1,ij} = \frac{\sum_{r=2}^{nk}\left[k_r\ \mathrm{Im}(w_{ij}\bar{a}_{ij})_{k=k_r}\right] - \sum_{l=1}^{nl} u_i d_{il}\left\{\sum_{r=2}^{nk}\left[\gamma_l k_r^2\big/\left(k_r^2+\gamma_l^2\right)\right]\right\}v_j e_{lj}}{\sum_{r=2}^{nk} k_r^2} \tag{15}$$

and

$$w_{ij}a_{2,ij} = \frac{\sum_{r=2}^{nk}\left[-k_r^2\ \mathrm{Re}(w_{ij}\bar{a}_{ij})_{k=k_r}\right] + \sum_{l=1}^{nl} u_i d_{il}\left\{\sum_{r=2}^{nk}\left[k_r^4\big/\left(k_r^2+\gamma_l^2\right)\right]\right\}v_j e_{lj}}{\sum_{r=2}^{nk} k_r^4} \tag{16}$$

It follows that if we scale matrix $\boldsymbol{A}$ and denote it by $\boldsymbol{A}^s$, and if we similarly denote the resulting scaled $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ matrices by the superscript s, then Eqs. (14–16) can be written as

$$F = \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk}\Bigg[\mathrm{Re}\left(\bar{a}_{ij}^s\right)_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{k_r^2}{\left(k_r^2+\gamma_l^2\right)}\bar{e}_{lj} + k_r^2 a_{2,ij}^s\Bigg]^2 + \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk}\Bigg[\mathrm{Im}\left(\bar{a}_{ij}^s\right)_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{\gamma_l k_r}{\left(k_r^2+\gamma_l^2\right)}\bar{e}_{lj} - k_r a_{1,ij}^s\Bigg]^2 \tag{17}$$

where

$$\bar{d}_{il} = u_i d_{il}, \qquad \bar{e}_{lj} = v_j e_{lj} \tag{18}$$

and

$$a_{1,ij}^s = \frac{\sum_{r=2}^{nk}\left[k_r \mathrm{Im}\left(\bar{a}_{ij}^s\right)_{k=k_r}\right] - \sum_{l=1}^{nl}\bar{d}_{il}\left\{\sum_{r=2}^{nk}\left[\gamma_l k_r^2\big/\left(k_r^2+\gamma_l^2\right)\right]\right\}\bar{e}_{lj}}{\sum_{r=2}^{nk} k_r^2} \tag{19}$$

$$a^s_{2,ij} = \frac{\sum_{r=2}^{nk}\left[-k_r^2 \mathrm{Re}\left(\bar{a}^s_{ij}\right)_{k=k_r}\right] + \sum_{l=1}^{nl} \bar{d}_{il}\left\{\sum_{r=2}^{nk}\left[k_r^4/\left(k_r^2+\gamma_l^2\right)\right]\right\}\bar{e}_{lj}}{\sum_{r=2}^{nk} k_r^4} \tag{20}$$

Referring back to the example given earlier whereby the weighting w_{ij} is very small and of order ε, Eqs. (17) and (18) show that a small change in the $\bar{d}_{il}$ and $\bar{e}_{lj}$ design variables is equivalent to large changes in d_{il} and e_{lj} design variables and thus lead to a significantly more rapid convergence rate.

For completeness, the expressions for the derivatives of the scaled matrices are presented in the following

$$\frac{\partial F}{\partial \bar{d}_{il}} = 2\sum_{j=1}^{m}\sum_{r=2}^{nk}\left[\mathrm{Re}\left(\bar{a}^s_{ij}\right)_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{k_r^2}{k_r^2+\gamma_l^2}\bar{e}_{lj} + k_r^2 a^s_{2,ij}\right]$$
$$\times\left[-\frac{k_r^2}{k_r^2+\gamma_l^2} + \frac{k_r^2}{\sum_{r=2}^{nk}k_r^4}\left(\sum_{r=2}^{nk}\frac{k_r^4}{k_r^2+\gamma_l^2}\right)\right]\bar{e}_{lj}$$
$$+2\sum_{j=1}^{m}\sum_{r=2}^{nk}\left[\mathrm{Im}\left(\bar{a}^s_{ij}\right)_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{k_r\gamma_l}{k_r^2+\gamma_l^2}\bar{e}_{lj} - k_r a^s_{1,ij}\right]$$
$$\times\left[-\frac{k_r\gamma_l}{k_r^2+\gamma_l^2} + \frac{k_r}{\sum_{r=2}^{nk}k_r^2}\left(\sum_{r=2}^{nk}\frac{k_r^2\gamma_l}{k_r^2+\gamma_l^2}\right)\right]\bar{e}_{lj} \tag{21}$$

$$\frac{\partial F}{\partial \bar{e}_{lj}} = 2\sum_{i=1}^{n}\sum_{r=2}^{nk}\left[\mathrm{Re}\left(\bar{a}^s_{ij}\right)_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{k_r^2}{k_r^2+\gamma_l^2}\bar{e}_{lj} + k_r^2 a^s_{2,ij}\right]$$
$$\times\left[-\frac{k_r^2}{k_r^2+\gamma_l^2} + \frac{k_r^2}{\sum_{r=2}^{nk}k_r^4}\left(\sum_{r=2}^{nk}\frac{k_r^4}{k_r^2+\gamma_l^2}\right)\right]\bar{d}_{il}$$
$$+2\sum_{i=1}^{n}\sum_{r=2}^{nk}\left[\mathrm{Im}(\bar{a}_{ij})_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{k_r\gamma_l}{k_r^2+\gamma_l^2}\bar{e}_{lj} - k_r a^s_{1,ij}\right]$$
$$\times\left[-\frac{k_r\gamma_l}{k_r^2+\gamma_l^2} + \frac{k_r}{\sum_{r=2}^{nk}k_r^2}\left(\sum_{r=2}^{nk}\frac{k_r^2\gamma_l}{k_r^2+\gamma_l^2}\right)\right]\bar{d}_{il} \tag{22}$$

and

$$\frac{\partial F}{\partial \gamma_l} = 2\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk}\left[\mathrm{Re}\left(\bar{a}^s_{ij}\right)_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{k_r^2}{k_r^2+\gamma_l^2}\bar{e}_{lj} + k_r^2 a^s_{2,ij}\right]$$
$$\times\left[\frac{2\gamma_l k_r^2}{\left(k_r^2+\gamma_l^2\right)^2} - \frac{k_r^2}{\sum_{r=2}^{nk}k_r^4}\left(\sum_{r=2}^{nk}\frac{2\gamma_l k_r^4}{\left(k_r^2+\gamma_l^2\right)^2}\right)\right]\bar{d}_{il}\bar{e}_{lj}$$
$$+2\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{r=2}^{nk}\left[\mathrm{Im}\left(\bar{a}^s_{ij}\right)_{k=k_r} - \sum_{l=1}^{nl}\bar{d}_{il}\frac{k_r\gamma_l}{k_r^2+\gamma_l^2}\bar{e}_{lj} - k_r a^s_{1,ij}\right]$$
$$\times\left[\frac{2k_r\gamma_l^2}{\left(k_r^2+\gamma_l^2\right)^2} - \frac{k_r}{k_r^2+\gamma_l^2} - \frac{k_r}{\sum_{r=2}^{nk}k_r^2}\right.$$
$$\left.\times\sum_{r=2}^{nk}\left(\frac{2k_r^2\gamma_l^2}{\left(k_r^2+\gamma_l^2\right)^2} - \frac{k_r^2}{k_r^2+\gamma_l^2}\right)\right]\bar{d}_{il}\bar{e}_{lj} \tag{23}$$

Clearly, once convergence is obtained, a return to the unscaled matrices is needed, using the following equations:

$$a_{1,ij} = a^s_{1,ij} \times colmax(j) \times rowmax(i)$$
$$a_{2,ij} = a^s_{2,ij} \times colmax(j) \times rowmax(i)$$
$$d_{ij} = \bar{d}_{ij} \times rowmax(i), \qquad e_{ij} = \bar{e}_{ij} \times colmax(j) \tag{24}$$

Numerical Example

Matrix $\boldsymbol{A}$ for the DAST-ARW1 wing will be taken as a numerical example. It consists of two rigid-body modes, eight-structural modes, one control surface coupling mode (coupling effects only), and one gust coupling mode (coupling effects only). Thus, the aerodynamic matrix is of order 10×12. In general, gust coupling involves relatively large aerodynamic terms. The numerical values of matrix $\boldsymbol{A}$ are given at 10 different values of k (0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8). The weighting used throughout this work will be determined using Eq. (12) at $k_f = 0.1$. An optimization example will be run using this weighting as a general type weighting and employing Eqs. (5–11). Following this, the very same example will be run using the same weighting, but taking advantage of the scaled formulation, using Eqs. (17–24). This is done in order to test the difference in the rates of convergence. In parallel, the conventional MS algorithm (similarly unconstrained) will be run for the following two cases:

1) Weighting is considered as a general weighting, thus leading to the determination of $\boldsymbol{E}$ and $\boldsymbol{D}$ a column and row at a time, respectively.

2) Weighting is considered as scaling weights, thus enabling the determination of the complete $\boldsymbol{E}$ and $\boldsymbol{D}$ matrices during a single LS analysis (instead of single columns and single rows during each single LS analysis).

Comparisons between the different runs will permit the showing of the effects of the new formulation on F, which is effectively unconstrained and does not use preassumed fixed values for the aerodynamic roots γ_l. The preceding comparisons will also show the effects of scaling vs weighting for both MS and the suggested optimization method, including the required CPU time. Finally, results using Roger's approximation, as applied to the scaled matrix $\boldsymbol{A}$, will also be presented for purpose of comparison.

Because explicit expressions were obtained for all of the elements of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ in the optimization method proposed herein, no computational advantages are obtained if equality constraints are imposed. Therefore, only a minimum number of essential equality constraints need be introduced, like the constraint at $k = 0$ used in this work. This is true because constraints in general are known to adversely affect the quality of the MS approximation.

The optimization method requires initial values for $\boldsymbol{D}$, $\boldsymbol{E}$, and $\boldsymbol{\gamma}$. Initially, the singular value analysis of Ref. 8 was applied to Roger's $\boldsymbol{A}_3$ matrix to provide the initial values for $\boldsymbol{D}$ and $\boldsymbol{E}$ matrices. However, it was later found that using the MS initial values for $\boldsymbol{D}$, and applying a similar MS procedure for $\boldsymbol{E}$, yielded equally good results for the scaled formulation. For the weighted formulation, an identical procedure was applied; however, the 1 were replaced by $rowmax(i)$ for $\boldsymbol{D}$ and by $colmax(j)$ for $\boldsymbol{E}$. All of the results to be presented in this work were obtained using these latter initial values for $\boldsymbol{D}$ and $\boldsymbol{E}$. The initial value for $\boldsymbol{\gamma}$ was taken to be the same in all methods (Roger's, MS, and optimization). These will be indicated in the tabulated results. All of the preceding examples were run on a laptop PC having Intel's Centrino Processor running at 1.6 GHZ.

Presentation and Discussion of Results

Tables 1 and 2 show the values for both $colmax(j)$ and $rowmax(i)$. It can be seen that the gust coupling mode (associated with column 12) involves large absolute values for the aerodynamic forces, which are at least one order of magnitude larger than the other modes. Inspection of $rowmax(i)$ indicates that the third and eighth rows have very small absolute values that are four and three orders of magnitude smaller, respectively, than most of the other rows. Because the weights are formed by the products of the reciprocals of $colmax(j)$ and $rowmax(i)$, they will vary between 6,520.2 ($w_{3,3}$) and 0.0002578 ($w_{1,12}$). Also, the maximum ratio between the preceding weights is of the order10^7! It is therefore evident that the elements associated with the third and eighth rows will dominate the changes in the values of the design variables during the optimization, whereas the elements associated with the twelfth column will change very slowly, thus significantly slowing down the

Table 1 Scaling of the columns of matrix *A*

j	1	2	3	4	5	6	7	8	9	10	11	12
$colmax(j)$	37.07	355.38	0.3	201.67	323.33	197.39	222.01	0.32	343.75	131.71	161.78	3878.37
v_j	0.0027	0.0028	3.33	0.00496	0.0031	0.00506	0.0045	3.12	0.0029	0.0076	0.0062	0.00026

Table 2 Scaling of the rows of matrix *A*

i	1	2	3	4	5	6	7	8	9	10
$rowmax(i)$	1.00	1.00	0.00051	0.16	1.00	0.51	1.00	0.0021	1.00	1.00
u_i	1.00	1.00	1960.78	6.25	1.00	1.96	1.00	476.19	1.00	1.00

Table 3 Convergence of weighted formulation vs scaled formulation (*nl* = 6), using optimization

$nitmax$	0	1	2	3	4	5	6	8	10	12	14	16	18	20	40
$\sqrt{F_{\text{weighted}}}$	4.223	1.556	1.110	0.884	0.829	0.785	0.770	0.716	0.662	0.617	0.596	0.578	0.565	0.547	0.518
$\sqrt{F_{\text{scaled}}}$	4.223	0.635	0.531	0.509	——	——	——	——	——	——	——	——	——	——	——

Table 4 Comparison of unconstrained results using Roger's, MS, and optimization methods

nl	1	2	3	4	5	6	7	8	9	10
Roger										
$\sqrt{F}$	2.0989	1.0959	0.7598	0.42028	0.16179	0.10766	0.0665	0.0460	0.03438	0.02350
rer	0.00530	0.00327	0.00201	0.00123	0.000662	0.000404	0.000299	0.000202	0.0001284	0.0001169
γ_l[a]	0.1	0.1, 0.2	0.1, 0.2, 0.3	0.1, 0.2 0.3, 0.4	0.1, 0.2, 0.3 0.4, 0.5	0.1, 0.2, 0.3 0.4, 0.5, 0.6	0.1, 0.2, 0.3 0.4, 0.5 0.6, 0.7	0.1, 0.2, 0.3 0.4, 0.5, 0.6 0.7, 0.8	0.1, 0.2, 0.3 0.4, 0.5, 0.6 0.7, 0.8, 0.9	0.1, 0.2, 0.3 0.4, 0.5, 0.6 0.7, 0.8 0.9, 1.0
Scaled opt. $nitmax = 2$										
$\sqrt{F}$	1.8987	1.4691	1.1211	0.8943	0.6849	0.5313	0.4141	0.3552	0.2789	0.2521
rer	0.00496	0.00521	0.00416	0.00385	0.00327	0.00222	0.00207	0.00180	0.00155	0.00153
γ_l[a]	0.220	0.298 0.412	0.239 0.615 0.897	0.145, 0.317 0.548, 0.843 0.938	0.178, 0.432 0.493, 0.600 0.874	0.128, 0.433 0.463, 0.574 0.575, 0.851	0.165, 0.381 0.464, 0.474 0.517, 0.546 0.577	0.167, 0.277 0.298, 0.487 0.491, 0.492 0.588, 1.209	0.150, 0.163 0.305, 0.423 0.516, 0.584 0.622, 0.722 0.799	0.155, 0.286 0.344, 0.422 0.628, 0.657 0.722, 0.724 0.751, 0.968
CPU, s	0.183	0.235	0.439	0.938	1.088	2.004	2.576	3.127	4.079	5.289
Scaled opt. $nitmax = 3$										
$\sqrt{F}$	1.8978	1.4689	1.1203	0.8801	0.6813	0.5094	0.4095	0.3152	0.2589	0.2365
rer	0.00499	0.00522	0.00415	0.00383	0.00326	0.00221	0.00200	0.00152	0.00152	0.00131
γ_l[a]	0.220	0.314 0.383	0.237, 0.672 0.791	0.160, 0.407 0.522, 0.708	0.179, 0.461 0.502, 0.582 0.757	0.145, 0.464 0.491, 0.561 0.567, 0.683	0.182, 0.411 0.466, 0.474, 0.493, 0.508 0.532	0.173, 0.261 0.298, 0.407 0.422, 0.469 0.544, 0.956	0.189, 0.200 0.325, 0.424 0.523, 0.613 0.636, 0.711 0.761	0.146, 0.317 0.358, 0.407 0.619, 0.637 0.680, 0.684 0.692, 0.818
CPU, s	0.161	0.350	0.678	1.022	1.650	2.347	3.282	4.396	5.629	7.426
MS NIT = 50										
$\sqrt{F}$	2.3760	1.6603	1.4524	1.0930	0.9359	0.7579	0.5207	0.3971	0.3053	0.2738
rer	0.00606	0.00482	0.00461	0.00384	0.00359	0.00314	0.00238	0.00224	0.00177	0.00153
Weighted CPU, s	0.103	0.159	0.236	0.306	0.380	0.513	0.638	0.966	0.924	1.052
Scaled CPU, s	0.0690	0.0810	0.0860	0.091	0.129	0.123	0.157	0.154	0.187	0.243
MS NIT = 200										
$\sqrt{F}$	2.3760	1.6603	1.4520	1.0396	0.8731	0.5950	0.5083	0.3498	0.2924	0.2572
rer	0.00606	0.00482	0.00465	0.00388	0.00381	0.00272	0.00228	0.00203	0.00168	0.00150
Weighted CPU, s	0.272	0.462	0.746	1.240	1.426	1.877	2.324	3.131	3.577	4.319
Scaled CPU, s	0.126	0.177	0.226	0.263	0383	0.391	0.434	0.501	0.600	0.652

[a] γ_l for MS are the same as Roger's.

convergence of the weighted optimization scheme. At this point, one could have cited the logical benefits obtained during the optimization process if the $w_{ijk_r}^2$ term in Eqs. (5–11) would have been replaced by w_{ijk_r}. However, this will not be done in this work because we wish to compare results with identical methods of weights as those currently in use.

Table 3 shows the rate of convergence of the optimization scheme vs *nitmax*, where number of iterations $= nitmax \times$ number of design variables. Two cases are shown for comparison: the weighted formulation against the scaled formulation for the case where $nl = 6$. (It should be remembered that the numerical values of the weights are identical in both cases.) Table 3 shows that the rate of convergence of the scaled formulation is about 10 times faster than the rate of the weighted formulation. Hence, the scaled formulation will be used throughout the rest of this work. Table 4 shows a comparison between the results obtained using the Roger approximation, the scaled optimization formulation, and the conventional MS formulation for the unconstrained case (except for $\boldsymbol{A}_0$). The scaled formulation results are presented for *nitmax* = 2 and 3. The MS results are given for two different values of *nit* $\boldsymbol{D} \rightarrow \boldsymbol{E} \rightarrow \boldsymbol{D}$ iterations. The results obtained using *nit* = 50 can be compared against those using *nit* = 200. These two numbers of iterations are presented in order to give the reader an indication regarding the convergence of the results presented. The added states in the MS and and the newly proposed optimization formulation are equal to the number of lag terms nl, whereas in the Roger approximation the number of added states is equal to $n \times nl$. This means that in our example $10 \times nl$ states are added by the Roger approximation. Both the Roger and the MS approximations require preassumed values for the lag terms in the γ matrix. The values used in the current example are also shown in the preceding table. Finally, as already stated earlier, the use of scaling weights has also a direct beneficial effect on the MS formulation because it permits the solving of the full $\boldsymbol{E}$ and $\boldsymbol{D}$ matrices in a single LS analysis (rather than solving for a single column of $\boldsymbol{E}$ and a single row of $\boldsymbol{D}$ in the conventional MS analysis). Table 4 shows results for both of these cases. These latter results are referred to in Table 4 as the scaled MS vs the weighted MS cases. Clearly, the residual errors in these two MS cases are identical; however, the CPU time needed to obtain the scaled MS solution is significantly reduced.

In all cases shown in Table 4, the inclusion of the aerodynamic lag terms as design variables (in the newly proposed optimization formulation) yields a better aerodynamic approximation and leads to lower residual errors. The difference between the MS results and the optimization results is more significant when a small number of lag terms is used. However, this improvement in optimized results can be offset by increasing by 1 the number of lag terms used by the standard MS method. For $nl = 1$, the optimization results are even better than those obtained using Roger's approximation (which leads to 10 augmented states)!!! The reason clearly lies in the fact that the optimization formulation permits changing the value of γ_1, which in this case has a relatively significant effect.

Table 4 also shows the optimal values of the γ_l terms. It is difficult to observe a clear pattern for these values, except that in some cases these lag terms tend to form a group with very close numerical values. For example, in the case where $nl = 7$, six of the seven lag terms lie within the range of 0.411 and 0.532!!! This phenomenon of near coalescence has been previously observed by Eversman and Tewari[9] while trying to determine optimal γ using a nongradient optimization method.

It is difficult to give exact comparative values regarding the computational labor required when using the different formulations because this clearly depends on the degree of the desired convergence. Nevertheless, it can be stated that, in general, the computational labor required by the optimization formulation is greater than the one required by the conventional MS method. However, the CPUs required are generally similar ! Also (in Table 4), if one adopts the scaling weights approach in the MS iterative LS method of solution, then the ratios between the scaled and the weighted CPUs are of the order of around 1:4 for the NIT = 50, and around 1:5 for the NIT = 200, with the CPU associated with the scaled weighting being much lower.

Finally, the parameter *rer* is also introduced in Table 4 in order to give an indication regarding the average relative error of the fitted approximation, where the parameter *rer* is defined (following Ref. 8) by

$$rer = \frac{1}{n \times m \times nk} \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{r=1}^{nk} \left\{ \text{cabs}\left[\frac{(\bar{a}_{ij})_{k=k_r} - ik_r a_{1,ij} + k_r^2 a_{2,ij} - ik_r \sum_{l=1}^{nl} d_{il}[1/(ik_r + \gamma_l)] e_{il}}{(a_{ij})_{k=k_r}} \right] \right\}^2} \tag{25}$$

It can be seen that in all of the cases shown in Table 4 the parameter *rer* assumes values that are equivalent to an average relative error that is smaller than 1%, thus indicating a good overall fit.

Conclusions

An efficient nonlinear optimization approach has been formulated to determine the aerodynamic matrices needed in the MS approximation. This optimization approach avoids the predetermination of the numerical values of the lag terms because they form an integral part of the design variables. As a result, a better fit of the aerodynamic matrix is obtained, leading to smaller residual errors. The formulation of the optimization method is based on the analytical computation of the gradients of the objective function F, as well as on the explicit formulation of expressions for some of the unknown matrices and on the appropriate scaling of the aerodynamic matrix. It is shown that general weights incorporated into the optimization method can significantly slow down the rate of convergence, and that scaling weights are beneficial for both accelerating the convergence and for counteracting the normalization effects during the computation of the generalized aerodynamic matrix $\boldsymbol{A}$. It is further shown that if scaling weights are used (thus leading to scaled matrices) the resulting computational labor is of the same order as the one needed using the current MS method of solution, which is based on preassumed aerodynamic roots. It is therefore clear that the method proposed in this work is far superior to any other optimization method used in the past that attempted to determine also the aerodynamic roots (and that were based on the MS double iterative LS method). It is also shown that the adoption of the scaling already mentioned in the current MS LS method of solution leads to a significant reduction in the computational labor involved. The example used in this work has yielded the surprising result that the optimization-based solution of the MS approximation with a single lag term (and thus a single augmented state) yields better results than the Roger approximation with one preassumed lag term (and thus leading to 10 augmented states).

It is believed that the scaling of the aerodynamic matrix should prove sufficient to produce good overall results in all cases (including counteracting the normalization effects of $\boldsymbol{A}$ mentioned earlier). However, if for any reason a deviation from these scaling weights is needed (such as in the case when physical weights are desired), it is recommended that such a deviation be applied to the *scaled* aerodynamic matrix, so as to avoid as much as possible the slow convergence associated with the use of general weights in the nonlinear optimization formulation.

References

[1]Roger, K. L., "Airplane Math Modeling Methods for Active Controls," AGARD, CP-228, Aug. 1977.

[2]Dunn, H. J., "An Analytical Technique for Approximating Unsteady Aerodynamics in the Time Domain," NASA TP-1738, Nov. 1980.

[3]Vepa, R., "Finite State Modeling of Aeroelastic Systems," NASA CR-2779, Feb. 1977.

[4]Karpel, M., "Design for Active Flutter Suppression and Gust Alleviation Using State-Space Aeroelastic Modeling," *Journal of Aircraft*, Vol. 19, No. 3, 1982, pp. 221–227.

[5]Karpel, M., "Time Domain Aeroservoelastic Modeling Using Weighted Unsteady Aerodynamic Forces," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, 1990, pp. 30–37.

[6]Karpel, M., and Strul, E., "Minimum State Unsteady Aerodynamic Approximations with Flexible Constraints," *Journal of Aircraft*, Vol. 33, No. 6, 1996, pp. 1190–1196.

[7]Tiffany, S. H., and Adams, W. M., "Non-Linear Programming Extensions to Rational Function Approximation Methods for Unsteady Aerodynamic Forces," NASA TP-2776, July 1988.

[8]Mor, M., and Livne, E., "State Space Unsteady Aerodynamics for Aeroservoelastic Configuration Shape Optimization of Flight Vehicles," AIAA Paper 2004-1762, April 2004.

[9]Eversman, W., and Tewari, A., "Consistent Rational Function Approximation for Unsteady Aerodynamics," *Journal of Aircraft*, Vol. 28, No. 9, 1991, pp. 545–552.