AIAA JOURNAL
Vol. 43, No. 10, October 2005

Approximation of Unsteady Aerodynamic Forces $Q(k, M)$ by Use of Fuzzy Techniques

Adrian Hiliuta* and Ruxandra Mihaela Botez†
Ecole de Technologie Supérieure, Montreal, Quebec H3C 1K3, Canada
and
Marty Brenner‡
NASA Dryden Flight Research Center, Edwards, California 93523

The unsteady aerodynamic forces acting on an F/A-18 aircraft are calculated in the frequency domain using a doublet-lattice method (subsonic regime) or a constant-pressure method (supersonic regime). To study the effects of the control laws on a flexible fly-by-wire aircraft structure and calculate the flutter velocities and frequencies, these aerodynamic forces must be approximated in the Laplace domain. We show here that in the case where the aerodynamic forces data are calculated for a range of unevenly spaced reduced frequencies, a combination of fuzzy clustering and shape-preserving techniques can be used to obtain a very good approximation of these unsteady aerodynamic forces. Because the approximation of these forces by this new method remains actually in the frequency domain, we could easily further use for their conversion from the frequency domain into Laplace domain, classical methods such as least squares or minimum state. Finally, care must be used in the choice of reduced frequencies range, to determine the method to be deployed.

Nomenclature

A_I	=	matrix with three elements from algorithms A_I
A_{II}	=	matrix with five elements from algorithm A_{II}
A_i	=	aerodynamic coefficients for the least-square approximation, $i = 0, 1, 2, 3, 4$
$\boldsymbol{C}$	=	damping matrix
$\bar{c}$	=	mean aerodynamic chord
j	=	imaginary number $= \sqrt{-1}$
$\boldsymbol{K}$	=	stiffness matrix
k	=	reduced frequency
M	=	Mach number
$\boldsymbol{M}$	=	mass matrix
$\boldsymbol{N}$	=	number of vibration modes
Q	=	unsteady aerodynamic matrix
Q_I	=	imaginary part of Q matrix
Q_R	=	real part of Q part
q_{dyn}	=	dynamic pressure
S_I, S_{II}	=	aerodynamic matrices for the least-squares approximation
V	=	true airspeed
β	=	aerodynamic lag terms
η	=	generalized coordinates

I. Introduction

AEROSERVOELASTICITY theory represents the combination of several theories regarding different aspects of aircraft dynamics. Aeroservoelastic interactions on aircraft are very complex problems to solve, and they are essential for an aircraft's certification. The instabilities that arise from the adverse interactions between the flexible structure, the aerodynamic forces, and the control laws acting on the structure can appear at any time inside the flight envelope, and so we can state that aeroservoelastic interactions concern mainly the research field located at the intersection of three disciplines: aerodynamics, aeroelasticity, and servocontrols. The unsteady aerodynamic forces on the aircraft (from STARS[1]) are calculated for an F/A-18 at one Mach number, for a range of 10 reduced frequencies k by use of the doublet-lattice method (DLM) in the subsonic regime, and the constant-pressure method (CPM) in the supersonic regime.

One main aspect of aeroservoelasticity is the approximation of these unsteady aerodynamic forces $Q(k, M)$, where M is the Mach number and k is the reduced frequency.[2] Three classical methods have been used until now in the literature to approximate the unsteady generalized forces by rational functions from the frequency domain to the Laplace domain[3–6]: least square (LS), matrix padé, and minimum state (MS). To date, the approximation that yields the smallest-order time-domain state-space model is the MS method.[7]

Poirion[8] used several MS approximations, obtained for several fixed Mach numbers and a spline interpolation method for Mach-number dependence. Therefore, he computed the unsteady aerodynamic forces for any couple (k, M), where k is the reduced frequency and M is the Mach number. A new approach based on a precise Padé approximation used four order-reduction methods for the last term of the approximation, a term which could be seen as a transfer function of a linear system. The approximation error for this new method is 12–40 times less than for the MS method for the same number of augmented states and is dependent on the choice of the order-reduction method. However, this method remains very expensive in terms of computing time.[9]

In this paper, we show that for a specific range of 10 reduced frequencies, which are not evenly spaced, the classical LS method does not adequately approximate the unsteady aerodynamic forces from the frequency domain into the Laplace domain and, therefore, the flutter velocities. With regard to this deficiency, we have found, for the same range of reduced frequencies k, an excellent combination of fuzzy clustering techniques with a shape-preserving "pchip" method that provides a good approximation of the unsteady aerodynamic forces in the frequency domain.

Our main contribution is that a new method using fuzzy-set theories has been found for the approximation of unsteady aerodynamic forces in the frequency domain, and this new method can be successfully applied when initial unsteady aerodynamic forces are calculated for any range of unevenly spaced reduced frequencies.

Received 23 September 2004; revision received 10 June 2005; accepted for publication 11 June 2005.

*Postdoctoral Researcher, Department of Automated Manufacturing Engineering, 1100 Notre Dame West. Member AIAA.
†Professor, Department of Automated Manufacturing Engineering, 1100 Notre Dame West. Member AIAA.
‡Aerospace Engineer, MS 4840 D/RS.

II. Aircraft Equations of Motion

The basic aeroelastic equation of motion is[1]

$$\boldsymbol{M}\ddot{\eta} + \boldsymbol{C}\dot{\eta} + \boldsymbol{K}\eta + q_{\text{dyn}}\boldsymbol{Q}(k, M)\eta = 0 \tag{1}$$

where $\boldsymbol{Q}$ is the aerodynamic forces matrix, and η is the vector of generalized coordinates. The matrix $\boldsymbol{Q}$ is computed in STARS by the DLM method in the subsonic regime and by the CPM method in the supersonic regime. The matrix of aerodynamic forces is computed for a range of reduced frequencies k and Mach numbers M that depend on the aircraft true airspeed V and is written under the following form:

$$\boldsymbol{Q}(k, M) = \boldsymbol{Q}_R(k, M) + j\boldsymbol{Q}_I(k, M) \tag{2}$$

where $\boldsymbol{Q}_R$ is the real part of $\boldsymbol{Q}$ and $\boldsymbol{Q}_I$ is the imaginary part of $\boldsymbol{Q}$. We replace Eq. (2) in Eq. (1), so that we obtain the following:

$$\boldsymbol{M}\ddot{\eta} + \left[\boldsymbol{C} + \frac{\bar{c}}{2kV}q_{\text{dyn}}\boldsymbol{Q}_I(k, M)\right]\dot{\eta} + [\boldsymbol{K} + q_{\text{dyn}}\boldsymbol{Q}_R(k, M)]\eta = 0 \tag{3}$$

where no excitation force applies. From Eq. (3), the second derivative of generalized coordinates is further obtained:

$$\ddot{\eta} = -\boldsymbol{M}^{-1}\left[\boldsymbol{C} + \frac{\bar{c}}{2kV}q_{\text{dyn}}\boldsymbol{Q}_I(k, M)\right]\dot{\eta} - \boldsymbol{M}^{-1}[\boldsymbol{K} + q_{\text{dyn}}\boldsymbol{Q}_R(k, M)]\eta \tag{4}$$

We rearrange Eq. (4) under the following matrix form:

$$\begin{bmatrix}\ddot{\eta}\\ \dot{\eta}\end{bmatrix} = \begin{bmatrix} -\boldsymbol{M}^{-1}\left[\boldsymbol{C} + \frac{\bar{c}}{2kV}q_{\text{dyn}}\boldsymbol{Q}_I(k, M)\right] & -\boldsymbol{M}^{-1}[\boldsymbol{K} + q_{\text{dyn}}\boldsymbol{Q}_R(k, M)] \\ \boldsymbol{I} & 0 \end{bmatrix} \times \begin{bmatrix}\dot{\eta}\\ \eta\end{bmatrix} \tag{5}$$

Matrix equation (5) can be also written as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\cdot\boldsymbol{x} \tag{6}$$

where

$$\boldsymbol{x} = \begin{bmatrix}\dot{\eta}\\ \eta\end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} -\boldsymbol{M}^{-1}\left[\boldsymbol{C} + \frac{\bar{c}}{2kV}q_{\text{dyn}}\boldsymbol{Q}_I(k, M)\right] & -\boldsymbol{M}^{-1}[\boldsymbol{K} + q\boldsymbol{Q}_R(k, M)] \\ \boldsymbol{I} & 0 \end{bmatrix}$$

From Eq. (6), we obtain solutions for $\boldsymbol{x}$, for known values of aerodynamic forces $\boldsymbol{Q}_R$ and $\boldsymbol{Q}_I$. Flutter speeds and frequencies are calculated from the $\boldsymbol{A}$ matrix eigenvalues. The LS method is the most common method to approximate the unsteady aerodynamic forces, and we present this method in detail in the next section.

III. Two Approaches A_I and A_{II} Used in the LS Algorithm

The $\boldsymbol{Q}(k)$ matrix is calculated for one Mach number $M = 0.85$ on the F/A-18 at NASA Dryden Flight Research Center for a range of unevenly spaced reduced frequencies $k = 0.9\text{e}{-4}$; 0.001; 0.01; 0.02; 0.1; 0.2; 1; 1.5; 2; and 4 using the finite element software STARS.

These unsteady aerodynamic forces can be approximated with Padé polynomials by use of the LS method[10] as follows:

$$\boldsymbol{Q}(k) = \boldsymbol{A}_0 + jk\boldsymbol{A}_1 + (jk)^2\boldsymbol{A}_2 + \frac{jk}{jk + \beta_1}\boldsymbol{A}_3 + \frac{jk}{jk + \beta_2}\boldsymbol{A}_4 + \ldots \tag{7}$$

By separating the real and imaginary parts, the aerodynamic lag terms can be also written under the following form:

$$\frac{jk}{jk + \beta_i} = \frac{jk(\beta_i - jk)}{k^2 + \beta_i^2} = \frac{k^2}{k^2 + \beta_i^2} + j\frac{k\beta_i}{k^2 + \beta_i^2}$$

We separate the real and imaginary parts in Eq. (7) by taking into consideration only two lag terms β_1 and β_2:

$$\boldsymbol{Q}_R(k) = \boldsymbol{A}_0 - k^2\boldsymbol{A}_2 + \frac{k^2}{k^2 + \beta_1^2}\boldsymbol{A}_3 + \frac{k^2}{k^2 + \beta_2^2}\boldsymbol{A}_4$$

$$\boldsymbol{Q}_I(k) = k\boldsymbol{A}_1 + \frac{k\beta_1}{k^2 + \beta_1^2}\boldsymbol{A}_3 + \frac{k\beta_2}{k^2 + \beta_2^2}\boldsymbol{A}_4 \tag{8}$$

We next present two approaches: A_I and A_{II}. With A_I, the first reduced frequency to be considered k_1 is quite small. In our case, its value is $k_1 = 0.9\text{e}{-4}$. Therefore, we can make the approximation $k_1^2 \approx 0$, so that Eqs. (8) become

$$Q_R(k_1) = \boldsymbol{A}_0, \qquad \frac{Q_I(k_1)}{k_1} = \boldsymbol{A}_1 + \frac{\boldsymbol{A}_3}{\beta_1} + \frac{\boldsymbol{A}_4}{\beta_2} \tag{9}$$

From where we can obtain $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ as follows:

$$\boldsymbol{A}_0 = \boldsymbol{Q}_R(k_1), \qquad \boldsymbol{A}_1 = \frac{\boldsymbol{Q}_I(k_1)}{k_1} - \frac{\boldsymbol{A}_3}{\beta_1} - \frac{\boldsymbol{A}_4}{\beta_2} \tag{10}$$

From Eqs. (8),

$$\tilde{\boldsymbol{Q}}_R(k) = \boldsymbol{Q}_R(k) - \boldsymbol{A}_0 = -k^2\boldsymbol{A}_2 + \frac{k^2}{k^2 + \beta_1^2}\boldsymbol{A}_3 + \frac{k^2}{k^2 + \beta_2^2}\boldsymbol{A}_4$$

$$= \begin{bmatrix} -k^2\boldsymbol{I} & \frac{k^2}{k^2 + \beta_1^2}\boldsymbol{I} & \frac{k^2}{k^2 + \beta_2^2}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{A}_3\\ \boldsymbol{A}_4\end{bmatrix} = \boldsymbol{S}_{R1}\boldsymbol{A}_I \tag{11a}$$

$$\tilde{\boldsymbol{Q}}_I(k) = \frac{\boldsymbol{Q}_I(k)}{k} - \boldsymbol{A}_1 = \frac{\beta_1}{k^2 + \beta_1^2}\boldsymbol{A}_3 + \frac{\beta_2}{k^2 + \beta_2^2}\boldsymbol{A}_4$$

$$= \begin{bmatrix} 0 & \frac{\beta_1}{k^2 + \beta_1^2}\boldsymbol{I} & \frac{\beta_2}{k^2 + \beta_2^2}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{A}_3\\ \boldsymbol{A}_4\end{bmatrix} = \boldsymbol{S}_{I1}\boldsymbol{A}_I \tag{11b}$$

We introduce the notations $\tilde{\boldsymbol{Q}}_I = [\tilde{\boldsymbol{Q}}_R \quad \tilde{\boldsymbol{Q}}_I]^T$ and $\boldsymbol{S}_1 = [\boldsymbol{S}_{R1} \quad \boldsymbol{S}_{I1}]^T$. Further, Eqs. (11) will be written in the following manner:

$$\tilde{\boldsymbol{Q}}_I = \boldsymbol{S}_1\boldsymbol{A}_I \tag{12}$$

A second approach A_{II} can be taken into account when all terms given in Eqs. (8) are considered. In this case, Eqs. (8) can be rewritten in the following form:

$$\boldsymbol{Q}_R(k) = \begin{bmatrix}\boldsymbol{I} & 0 & -k^2\boldsymbol{I} & \frac{k^2}{k^2 + \beta_1^2}\boldsymbol{I} & \frac{k^2}{k^2 + \beta_2^2}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_0\\ \boldsymbol{A}_1\\ \boldsymbol{A}_2\\ \boldsymbol{A}_3\\ \boldsymbol{A}_4\end{bmatrix} = \boldsymbol{S}_{R2}\boldsymbol{A}_{II}$$

$$\boldsymbol{Q}_I(k) = \begin{bmatrix}0 & k\boldsymbol{I} & 0 & \frac{k\beta_1}{k^2 + \beta_1^2}\boldsymbol{I} & \frac{k\beta_2}{k^2 + \beta_2^2}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_0\\ \boldsymbol{A}_1\\ \boldsymbol{A}_2\\ \boldsymbol{A}_3\\ \boldsymbol{A}_4\end{bmatrix} = \boldsymbol{S}_{I2}\boldsymbol{A}_{II} \tag{13}$$

In this second approach, at least one column in $\boldsymbol{S}_{R2}$ or in $\boldsymbol{S}_{I2}$ is zero, and for this reason these two matrices cannot be inversed.

It is easier to introduce the following notations for the composed matrices including real and imaginary values:

$$\tilde{\boldsymbol{Q}}_{II} = [\boldsymbol{Q}_R \quad \boldsymbol{Q}_I]^T, \qquad \boldsymbol{S}_2 = [\boldsymbol{S}_{R2} \quad \boldsymbol{S}_{I2}]^T \tag{14}$$

so that Eqs. (13) become

$$\tilde{\boldsymbol{Q}}_{II} = \boldsymbol{S}_2 \boldsymbol{A}_{II} \tag{15}$$

In both approaches, the $\boldsymbol{A}$ matrices ($\boldsymbol{A}_I$ and $\boldsymbol{A}_{II}$) can be calculated by applying the LS method to Eqs. (12) and (15), respectively:

$$\boldsymbol{A} = (\boldsymbol{S}^T \boldsymbol{S})^{-1} \boldsymbol{S}^T \tilde{\boldsymbol{Q}} \tag{16}$$

The aerodynamic forces approximations in the frequency domain calculated with the second approach are closer to the original aerodynamic forces data than the ones calculated with the first approach, as the conditioning number is smaller. For this reason, we study different methods of interpolation for aerodynamic forces approximations in the frequency domain.

IV. $Q_{11}(k)$ Approximation by Use of Four MATLAB® Interpolation Methods

MATLAB software presents four main methods for data approximation or interpolation methods: "nearest," nearest neighbor interpolation; "linear," linear interpolation; "spline," cubic spline interpolation; and "pchip," piecewise cubic Hermite interpolation. However, we must specify that these four methods are not used for the approximation of unsteady aerodynamic forces from the frequency into the Laplace domain (as LS is used), but to approximate or interpolate the aerodynamic unsteady forces data in the frequency domain.

Each interpolation method might require comparatively more or less memory and/or longer computation time. Therefore, we need to take into consideration the memory and computation time required for each interpolation method. Nearest neighbor interpolation is the fastest method, but remains inconvenient as it provides the worst results in terms of smoothness. The linear interpolation method uses more memory than the nearest neighbor method and requires slightly more execution time. Unlike nearest neighbor interpolation, its results are continuous; however, the slope changes at the vertex points, which makes it difficult to be applied. The cubic spline interpolation has the longest relative execution time, although it requires less memory than cubic pchip interpolation. This method gives the smoothest results among all of the interpolation methods. According to the MATLAB documentation, unexpected results can however be obtained by the cubic spline interpolation method if the input data are not uniform, which is the situation here where we have nonuniform distributed data as a function of reduced frequency. The cubic pchip interpolation method requires more memory and execution time than either the nearest neighbor or linear methods. However, both the interpolated data and its derivative are continuous.

After using each of these four methods, we have concluded that the pchip method is the best method because it is less expensive to set up and has no overshoots and less oscillation than the other three methods, advantages that override its memory and execution time requirements. All of the methods in this section are of the interpolant type. Therefore, the fit residuals are always zero (within computer precision) because interpolants pass through the data points. The data to be interpolated are unevenly spaced, which means that we have data points for a total of six low reduced frequencies (smaller than 0.2), and other data points for a total of four higher reduced frequencies (higher than 0.2).

Figure 1 shows the approximation of one aerodynamic force element $Q_{11}(k)$ by use of four MATLAB methods: nearest and linear (upper Fig. 1) and spline and pchip (lower Fig. 1). We concluded from these types of tests that though the pchip method is the best method to be used in MATLAB for these aerodynamic forces interpolations, its results are not sufficiently accurate. To overcome this problem, the pchip method is combined with another method from fuzzy set theory, as shown in the next sections.

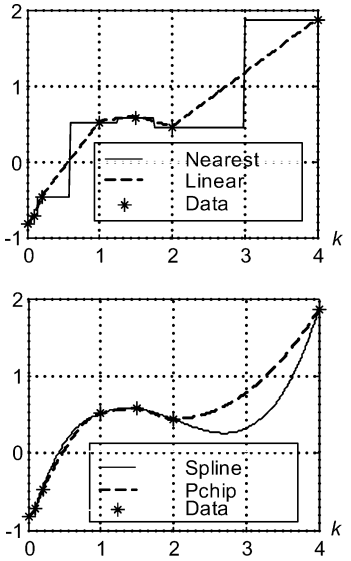


Fig. 1 Unsteady aerodynamic forces $Q_{11}(k)$ approximation by use of four MATLAB interpolation methods.

V. Use of Fuzzy Algorithms

The interpolation methods presented in the preceding section have their disadvantages, and, therefore, we have chosen to explore fuzzy algorithms for data approximations. There are two types of Sugeno-type adaptive neural fuzzy inference systems: one uses a grid partition of the data, and the other uses the subtractive clustering method. The adaptive neural fuzzy inference system ANFIS is generated by using one of these methods at a time, and then the same system is trained using a backpropagation algorithm.

The first method is the generation of an ANFIS structure from a data set, by use of a grid partition on the data, which means that the whole domain of data (two dimensional or three dimensional) is divided into evenly distributed partitions.

One membership function is allocated for each partition. We can specify the number of membership functions to be used (implicitly the number of partitions) as well as the type of output vs input memberships. The system is trained so that the rms error between the data and its approximation is minimized. The main disadvantage of the method is, once again, the nonuniformity of data distributed as a function of reduced frequencies, even if the initial grid partition space is uniform.

If we choose very small partitions (such as 0.01 in our case study), then a very large number of membership functions is used, but we will obtain many partitions without points. To clarify this statement, we will consider our case study, for which the reduced frequencies are 0.9e−4; 0.001; 0.01; 0.02; 0.1; 0.2; 1; 1.5; 2; and 4. We will fix a small partition—in this case, with an interval of 0.01. Therefore, from 0 to 0.01, we have three points; from 0.01 to 0.02 we have two points; from 0.02 to 0.03, from 0.03 to 0.04, from 0.04 to 0.05, from 0.05 to 0.06, from 0.06 to 0.07, from 0.07 to 0.08, and from 0.08 to 0.09 we will have zero points; from 0.09 to 1 we will have one point; from 1 to 1.01, . . . , from 1.48 to 1.49 we will have many intervals with zero points; from 1.49 to 1.5 we will have one point; but again we will have many intervals of zero points between 1.5 and 2 and between 2 and 4. The disadvantage of the choice of these small partitions is that a larger number of points in the first partitions are obtained than in the later partitions and we also have a very large number of partitions with zero points.

Let us assume that we have a partition with an interval length of 0.5. Therefore, its corresponding distribution will be as follows: from 0 to 0.5, six points; from 0.5 to 1, one point; from 1 to 1.5, two points; from 1.5 to 2, one point; from 2 to 2.5, one point; from 2.5 to 3, zero points; from 3 to 3.5, zero points; and from 3.5 to 4, one point. Then, we consider a partition with large intervals equal to 1 for the same type of reduced frequencies data. Therefore, in the interval [0, 1] we have seven points; in the interval [1, 2] there are

three points; in [2, 3] there is one point; and in [3, 4] one point. We can see that in all three cases, with partition intervals of 0.01, 0.5, and 1, the number of points vary, and sometimes there are no points in certain intervals (when we consider the partition interval of 0.5 or 0.01). We have shown that there is a disadvantage with this method when we want to use it on our model because of the nonuniform (unevenly) reduced frequencies range.

The second method in using fuzzy set theory is the generation of a fuzzy inference system (FIS) by use of the fuzzy subtractive clustering method. The identification of clusters of data from a large data set will produce a concise representation of the system's behavior. For every cluster found, one membership function is generated. The advantage of this approach is that we are able to work with unevenly distributed data. The rule extraction method first uses clustering to calculate the number of rules and antecedent membership functions and then uses a linear least-squares estimation to determine each rule's consequent equations. This function creates an FIS structure that contains a set of fuzzy rules that cover all of the featured space.

Training the system is realized by means of the hybrid method, with a least-squares fitting and backpropagation algorithm. After the FIS is generated, we train the systems to minimize the rms error. The results obtained by a trained FIS are presented in Figs. 2 and 3 with a solid line. The training process is long and difficult as it is based on the trial-and-error method. However, the results are saved in the computer memory as structures that contain the membership functions and the rules of fuzzy systems. These rules are logical sentences that model human reasoning. The most commonly used membership functions are Gaussian and are stored as a list of parameters (average and standard deviation) defining the shape of the function to be evaluated. Evaluation of the structure, after training, is a very easy process and does not consume much processor time. All membership functions are evaluated at the same time, and the output is obtained through a very simple operation because the output is the maximum of each of the activated membership functions. The advantage is that the training is done only once, for new data (in our case, a new aircraft), and only an evaluation is necessary afterwards.

Approximations of unsteady aerodynamic forces in both cases, gridpartition and clustering, are presented in Figs. 2 and 3 with dotted lines for untrained algorithms and solid lines for trained algorithms. After analyzing the approximations presented in Fig. 2, we can see that the generated system (untrained) provides us with an output always equal to zero. With training, the shape of the approximation is good, with the exception of the last two data points (where $k = 2$ and 4). The fact that the points (reduced frequencies k for which aerodynamic forces $\boldsymbol{Q}$ are calculated) are not evenly spaced, with more points in the beginning and less points in the end, affects the training algorithm in the grid-partition case. We can see that the solid line in Fig. 3 between data for $k = 2$ and 4 (representing the trained grid-partition algorithm results) is almost a straight line, whereas the solid line in Fig. 2 (representing the trained clustering ANFIS algorithm results) does an overfitting between data for $k = 2$ and 4. We concluded that the results of the approximations presented in Fig. 3 are much better than the results presented in Fig. 2, which gives a clear indication that the clustering algorithm is much better than the grid-partition algorithm.

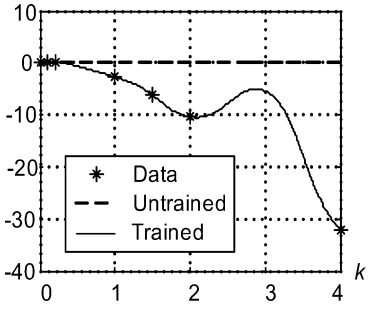


Fig. 2 Approximation of unsteady aerodynamic force $Q_{11}(k)$ by the grid-partition-type generated FIS algorithm, untrained and trained.

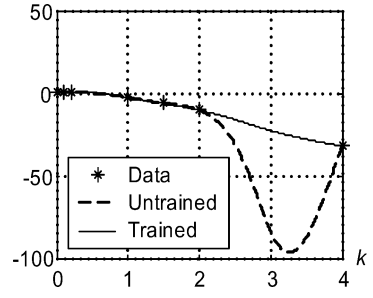


Fig. 3 Approximation of unsteady aerodynamic force $Q_{11}(k)$ by the clustering ANFIS-type algorithm, untrained and trained.

VI. Comparison of Different Interpolation Methods

The initial data are distributed as a function of reduced frequency $k = 0.9\text{e}{-4}$; 0.001; 0.01; 0.02; 0.1; 0.2; 1; 1.5; 2; and 4. Data are calculated uniformly in the interval $k = 0.9\text{e}{-4}$; 0.001; 0.01; 0.02; 0.1; 0.2, but they are calculated nonuniformly between 0.2 and 1, between 1 and 1.5, between 1.5 and 2, and between 2 and 4.

To overcome these large differences in reduced frequency intervals, we use the decimal logarithm of these parameters as follows: $\log_{10}(k) = -4.04$; -3; -2; -1.7; -1; -0.7; 0; 0.18; 0.30; 0.60. These modifications in data distribution cannot be applied in the least-squares algorithms. However, the new logarithmic distribution will affect the data approximation in all other cases such as spline, pchip, fuzzy grid, and fuzzy clustering. These new approximations are visible in Figs. 4–6. Among the MATLAB interpolation methods, we represent only spline and pchip approximations (see Fig. 4), as the other two methods of approximation, that is, nearest and linear, were already found to be not good enough. Figures 5 and 6 show the comparison of results obtained with the fuzzy grid partition vs the results obtained with the clustering algorithm. Figure 5 shows the untrained approximations, whereas Fig. 6 shows the trained approximations results.

VII. Smoothness of Approximated Function Derivatives

The approximations look more uniformly distributed in Figs. 4–6, where logarithmic scale is considered, than in preceding figures. To

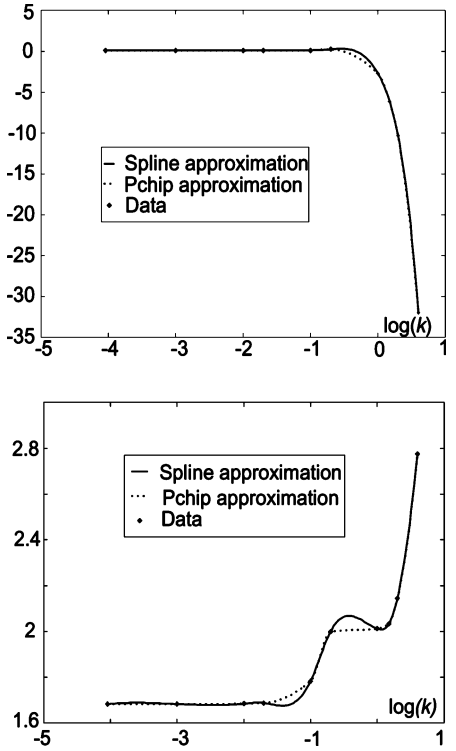


Fig. 4 MATLAB spline and pchip approximation methods for $Q_{11}(k)$ and $Q_{13}(k)$ with $\log_{10}(k)$.

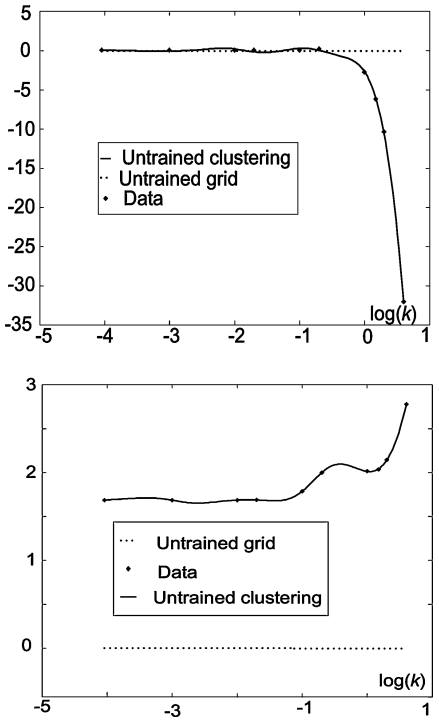


Fig. 5 Untrained fuzzy grid partition and clustering approximations for $Q_{11}(k)$ and $Q_{13}(k)$ with $\log_{10}(k)$.

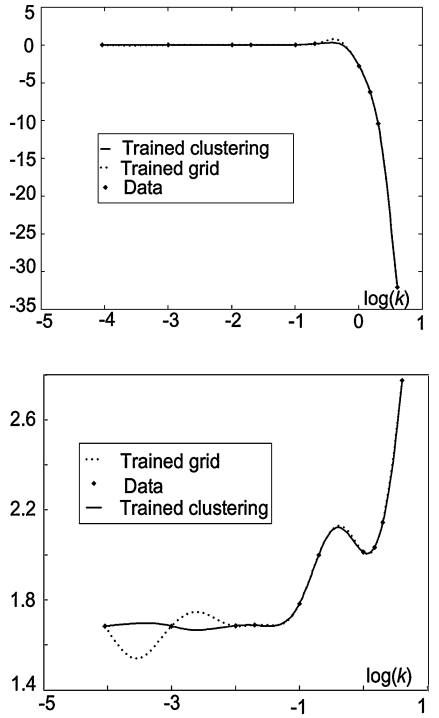


Fig. 6 Trained fuzzy grid-partition and clustering approximations for $Q_{11}(k)$ and $Q_{13}(k)$ with $\log_{10}(k)$.

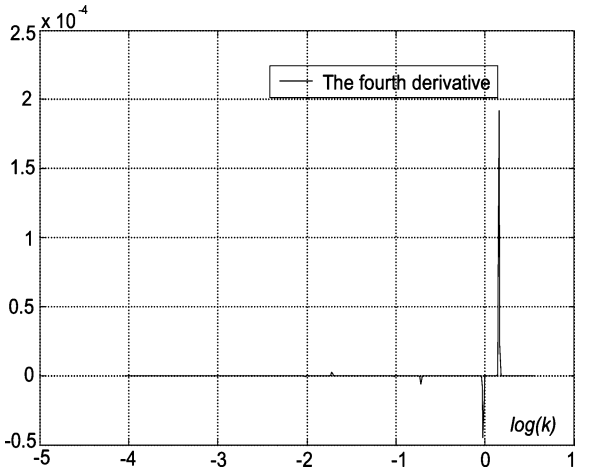


Fig. 7 Fourth derivative of spline approximation.

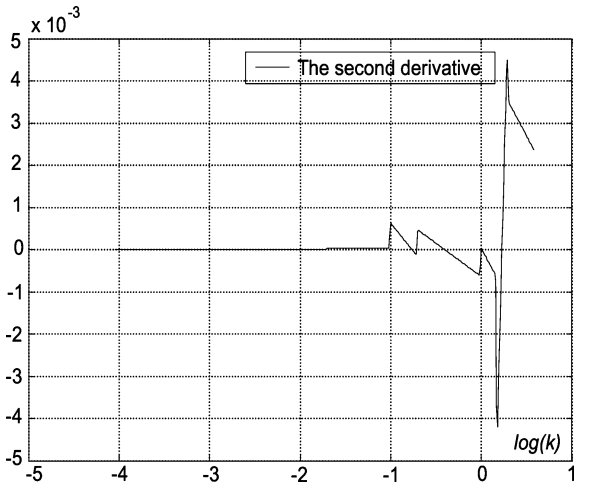


Fig. 8 Second derivative of pchip approximation.

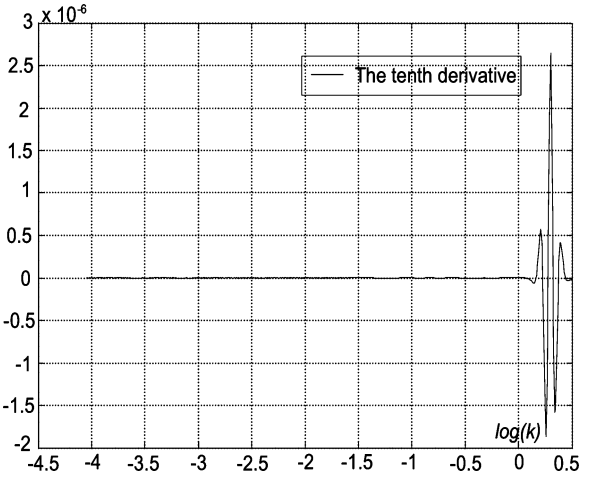


Fig. 9 Tenth derivative of fuzzy grid-partition approximation.

make a good decision regarding the scale to be used, we need to study the smoothness of the approximated functions, and we choose always the first aerodynamic matrix element $\boldsymbol{Q}_{11}$. The fact that we are approximating physical quantities leads us to think that their corresponding functions should be continuous, with no jumps, which means with no finite variation of the first derivative. To analyze the smoothness of the approximation function variations, we draw Figs. 7–10. We want to emphasize that we have depicted the maximum derivative that is obviously not continuous for each method considered in Figs. 4–6. We have determined (as seen in Figs. 9 and 10) that the approximated functions obtained with fuzzy algorithms (grid partition and clustering) are smooth even after 10 derivatives.

The best approximation, in terms of smoothness, is provided by the fuzzy clustering method. However, when there are insufficient data, the approximation is not good enough in all cases. To solve this problem, we use the approximation pchip, which is shape preserving, to increase the number of data points. By use of the pchip method, we generate new points in between original data: one new point in the middle of each interval. In this way, we combine the shape of pchip with the smooth results provided by the fuzzy

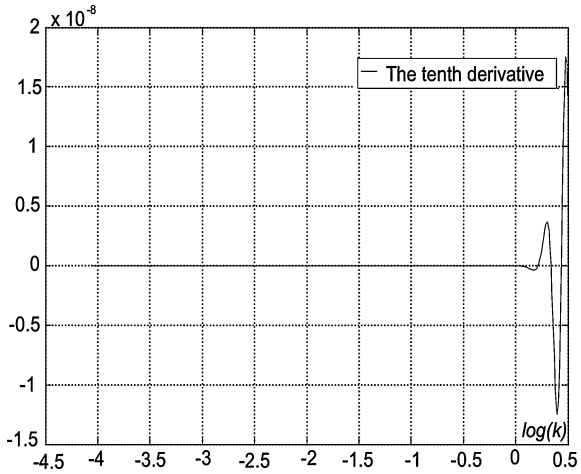
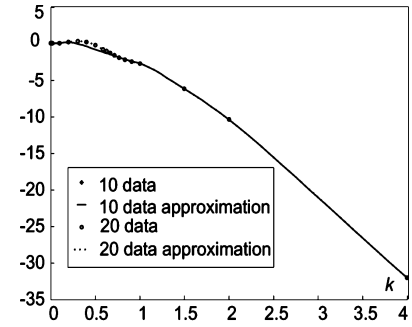


Fig. 10 Tenth derivative of fuzzy clustering approximation.



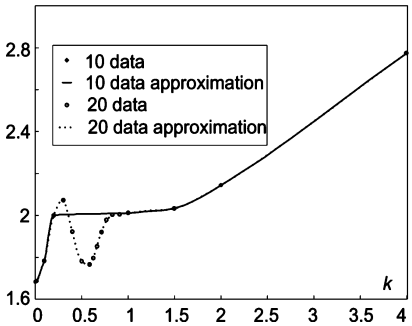


Fig. 11 Approximated data $Q_{11}(k)$ and $Q_{13}(k)$ by use of a combination of fuzzy clustering technique with shape-preserving pchip approximation with k for 10 and 20 sets of data points.

clustering approach. With the newly obtained points we can generate and train the fuzzy clustering system. Until now, we calculated the aerodynamic forces for 10 reduced frequencies. For comparison purposes, we choose a second range of 20 reduced frequencies $k =$ 9.09e−005; 0.001; 0.01; 0.02; 0.1; 0.2; 0.30; 0.4; 0.5; 0.59; 0.63; 0.67; 0.71;0.77;0.83;0.91;1; 1.5; 2; and 4. We denoted in Fig. 11 the results of the aerodynamic forces approximations for 10 k by 10 data and for 20 k by 20 data. From Fig. 11, we can see the aerodynamic forces differences between the results with 10 data vs 20 sets of data points. The conclusion drawn from Fig. 11 is that the number of k for which aerodynamic forces are approximated is very important (shape of the approximation changes with the number of points), and so it is important to have the maximum possible number of data.

The first and second derivatives calculated by use of pchip approximations are shown in Fig. 12, and the first and second derivatives calculated by use of the fuzzy clustering approximation are shown in Fig. 13.

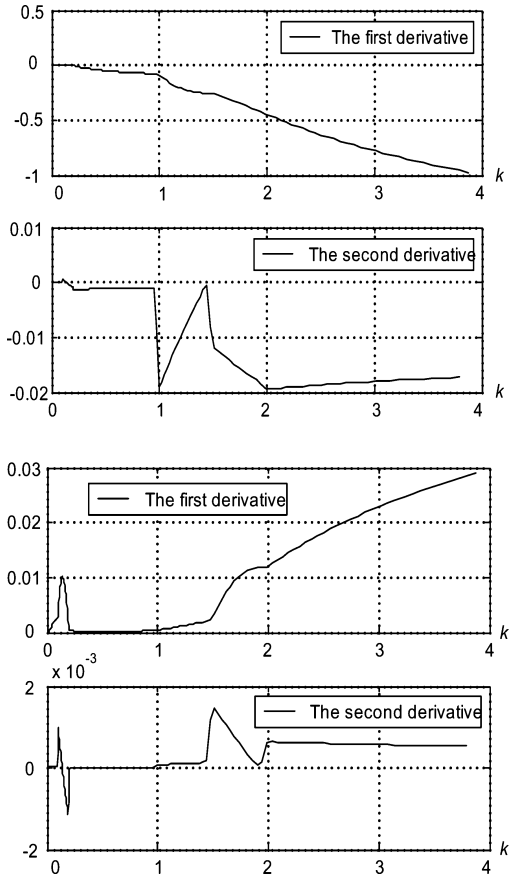


Fig. 12 First and second derivatives of approximations by using pchip interpolation, for the elements $Q_{11}(k)$ and $Q_{13}(k)$.

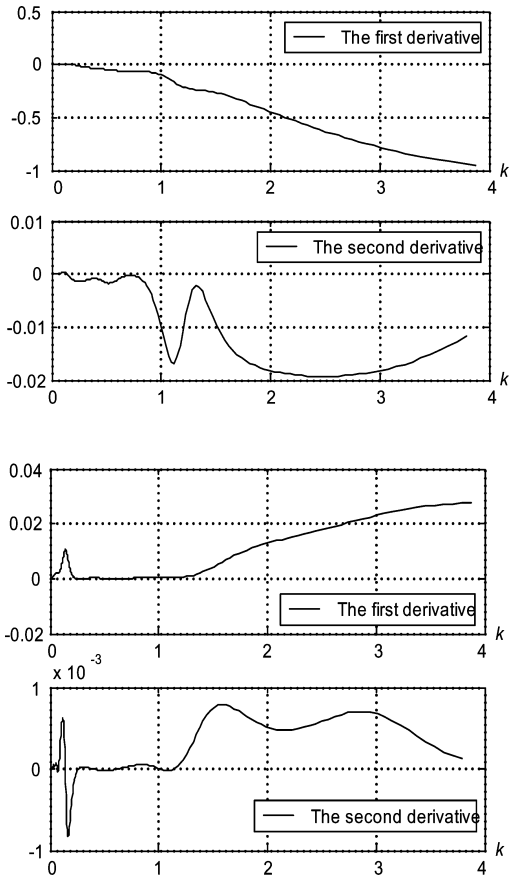


Fig. 13 First and second derivatives of approximations by using fuzzy clustering interpolation, for the elements $Q_{11}(k)$ and $Q_{13}(k)$.

We calculated the flutter speeds for the F/A-18 aircraft by use of 10 data points and 20 data points, which means by use of aerodynamic forces calculated for 10 reduced frequencies and 20 reduced frequencies. For example, the differences between the flutter speeds calculated with 10 data and the flutter speeds calculated with 20 data can be calculated by the LS method with the following equation:

$$V_F = \frac{V_{\text{LS 10 data}} - V_{\text{LS 20 data}}}{V_{\text{LS 10 data}}} * 100\% \tag{17}$$

The flutter speeds for our F/A-18 model by use of the LS method are 3.97% for V_{F1}, the first flutter speed, and −8.13% for V_{F2}, the second, flutter speed, for 10 and 20 data results comparison.

These differences in flutter speeds by use of 10 vs 20 data are high (almost 4% for the first flutter speed and 8% for the second flutter speed), and so it is shown that the number of reduced frequencies for which aerodynamic forces are calculated is very important. In addition, flutter speeds found by use of the LS with 20 data have same flutter speeds found by the pk flutter standard method, which means that the aerodynamic forces Q should be calculated for 20 reduced frequencies to give the best results.

VIII. Conclusions

Different interpolation methods were used for the unsteady aerodynamic forces approximations in the frequency domain, which were calculated for an unevenly range of reduced frequencies. These four methods were as follows: nearest, nearest neighbor interpolation; linear, linear interpolation; spline, cubic spline interpolation; and pchip, piecewise cubic Hermite interpolation. Each method has its advantages and disadvantages. The nearest method was fastest but inconvenient in terms of smoothness, and the linear method' was difficult to be applied as the slope changes at the vertex points. The spline method gave unexpected results if the input data were not uniform, which occured in the present case. The pchip interpolation method gave continuous data and derivatives. For this reason, among these four methods, the pchip method was selected as the best method.

We further explored two types of algorithms from the fuzzy theory. The first algorithm used the generation of an ANFIS structure from a data set, by use of data grid partition where the data were divided into evenly distributed partitions. We were unable to apply this method because of the nonuniformity of the reduced frequencies range. The second algorithm used the generation of an FIS by use of the fuzzy subtractive clustering method. We compared the results obtained by these two algorithms, and we found out that the clustering algorithm gave better results than the grid-partition algorithm even if the unsteady aerodynamic data were nonuniformly distributed.

Following the analysis done with fuzzy algorithms derivatives, we found out that the derivatives of these approximated unsteady aerodynamic forces functions obtained with both fuzzy algorithms (grid partition and clustering) were smooth after 10 derivatives.

Finally, the best combination of pchip and the fuzzy clustering technique fuzzy techniques was applied for aerodynamic unsteady forces calculated for a range of 10 unevenly spaced reduced frequencies $k = 0.9\text{e}{-4}$; 0.001; 0.01; 0.02; 0.1; 0.2; 1; 1.5; 2; and 4. If the range of reduced frequencies were chosen evenly, for 20 evenly spaced reduced frequencies $k = 9.09\text{e}{-5}$; 0.001; 0.01; 0.02; 0.1; 0.2; 0.30; 0.4; 0.5; 0.59; 0.63; 0.67; 0.71; 0.77; 0.83; 0.91; 1; 1.5; 2; and 4, we obtained very good results by using the least-square method, in which case we did not need to use the fuzzy techniques.

Differences up to 8% were found in flutter speeds calculated for the unsteady aerodynamic forces in the unevenly range of 10 reduced frequencies with respect to standard flutter speeds. For aeroservoelasticity studies, there is still a need to apply a classical method such as the LS or the MS methods over the aerodynamic forces interpolated by use of our new method to obtain the best aerodynamic forces approximations from frequency to Laplace domain.

Acknowledgments

The authors thank Kajal Gupta of the NASA Dryden Research Flight Center for permission to use the ATM in STARS. Thanks are also given other members of the STARS Engineering Group for their continuous assistance and collaboration: Tim Doyle and Shun Lung.

References

[1]Gupta, K. K., "STARS—An Integrated, Multidisciplinary, Finite-Element, Structural, Fluids, Aeroelastic, and Aeroservoelastic Analysis Computer Program," NASA TM-4795, May 1997.

[2]Rodden, W. P., Harder, R. L., Bellinger, E. D., "Aeroelastic Addition to NASTRAN," NASA CR-3094, March 1979.

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

[4]Edwards, J. W., "Unsteady Aerodynamic Modeling and Active Aeroelastic Control," Stanford Univ., SUDAAR 504, Stanford, CA, Feb. 1977.

[5]Roger, K. L., "Airplane Math Modeling Methods for Active Control Design," *Structural Aspects of Active Controls*, AGARD CP-228, Aug. 1977, pp. 4.1–4.11.

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

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

[8]Poirion, F., "Multi-Mach Rational Approximation to Generalized Aerodynamic Forces," *Journal of Aircraft*, Vol. 33, No. 6, 1996, pp. 1199–1201.

[9]Cotoi, I., and Botez, R. M., "Method of Unsteady Aerodynamic Forces Approximation for Aeroservoelastic Interactions," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, 2002, pp. 985–987.

[10]Lind, R., and Brenner, M., *Robust Aeroelastic Stability Analysis*, Springer-Verlag, London, 1999.

B. Balachandran
Associate Editor