

CFD-BASED AEROSERVOELASTIC PREDICTIONS WITH COMPARISONS TO BENCHMARK EXPERIMENTAL DATA

Cole H. Stephens* and Andrew S. Arena, Jr.†
School of Mechanical and Aerospace Engineering
Oklahoma State University
Stillwater, OK 74078
 and
 Kajal K. Gupta‡
NASA Dryden Flight Research Center
Edwards, CA 93523

Abstract

This study demonstrates the use of an Euler-based, unsteady CFD flow solver to model the aeroservoelastic behavior of the Benchmark Active Controls Technology (BACT) wing. For the steady case, results illustrate that a control surface deflection as large as 5° may be modeled without loss of accuracy due to viscous effects. Secondly, the unsteady characteristics of a benchmark wing are addressed showing that flutter prediction agrees well with experimental data over a wide Mach number range. Finally, as a combination of the latter, the development and implementation of a control law into the CFD-based aeroservoelastic model for the purpose of flutter suppression is addressed.

Nomenclature

CFD	Computational Fluid Dynamics
ASE	Aeroservoelasticity
M	Mach Number
C_P	Pressure Coefficient
CG	Center of Gravity
EA	Elastic Axis
a	Angle of Attack (deg)
r	Free-Stream Density (psf)
q_F	Dynamic Pressure at Flutter (psf)
K_H	Plunge Stiffness (lb/in)
K_a	Pitch Stiffness (in-lb/rad)
x_{cg}	Relative Location of the Elastic Axis to the Center of Gravity (Positive Aft) (in)
d	Control Surface Deflection (deg)
d_d	Desired Control Surface Deflection (deg)
K₁	Plunge Control Gain (deg/in)
K₂	Plunge-Rate Control Gain (deg/in/s)
K₃	Pitch Control Gain (deg/deg)

* Graduate Research Assistant, Student Member AIAA

† Associate Professor, Department of Mechanical and Aerospace Engineering, Senior Member AIAA.

‡ Aerospace Engineer, Member AIAA.

Copyright © 1999 by Cole H. Stephens and Andrew S. Arena, Jr. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

K₄	Pitch-Rate Control Gain (deg/deg/s)
k	Actuator Model Gain (1.02 deg/deg)
z	Actuator Model Damping (0.56)
w	Actuator Model 2 nd Order Frequency (165.3 rad/sec)

Introduction

Continuing advances in computational speed and memory have facilitated the use of CFD-based aeroelastic and aeroservoelastic analysis within the application oriented environment. Combining the computational horsepower of contemporary workstations with the need for accurate predictions of an aircraft's aeroelastic tendencies, it is now more feasible than ever provide quality estimates of the behavior of an aircraft prior to flight-test operations. At the core of these estimates are aeroservoelastic simulations that model not only the unsteady aerodynamics, but also allow the controlled deflection of one or more control surfaces.

When looking at representative aeroservoelastic simulations, there are typically three types of analyses involved in the complete time-marched solution: structural dynamics, unsteady-aerodynamics, and controls. Typically, the amount of time required for the CFD portion of the simulation greatly exceeds the time necessary for either the structural dynamics or controls. Hence it is therefore important to provide an accurate CFD solution in a minimal amount of time if the solution is to be used in an operational environment.

For the typical aeroelastic simulation, one generally considers the relatively smooth, continuous deformations encountered during flutter. A variety of techniques are available to incorporate these changing displacement and velocity boundary conditions into the CFD solution¹⁻³. Looking more closely at the aeroservoelastic problem, one notices a unique aspect not present in typical aeroelastic analyses. For the case of a control surface incorporated into a wing, the very close proximity or discontinuous nature of the control surface with adjacent sections of the airframe substantially increases the difficulty involved in accounting for deflections of this control surface.

A moving control surface is, in essence, simply a change in CFD geometry. Incorporating this change, however slight, into a time-marched solution is a challenging issue. An obvious choice would be the re-definition of the CFD geometry to include the control surface deflection, also known as the re-meshing option. Dynamic meshes incorporating a spring network analogy have also been demonstrated as an effective means of accounting for surface deflections¹. Finally, the transpiration method has also demonstrated its effectiveness for modeling surface deformations without the need for a complete re-definition of the computational geometry^{2,3,4}.

The re-meshing approach is, perhaps, the most versatile option, but considering the time, both user and computational, involved in completely re-defining the CFD geometry at each time-step, this method, with current hardware, is still very computationally inefficient. Additionally, as the control surface is deflected, “new” surfaces are exposed and change with each new control surface deflection. Likewise, due to the close proximity of a control surface with adjacent sections of the wing, these narrow gaps generally represent difficult computational regions for CFD flow solvers.

The spring analogy incorporated into a dynamic mesh also encounters difficulty in the regions involved with a control surface deflection. In these regions, the mesh must essentially be sheared at the intersecting regions. This shearing can cause significant loss of mesh-refinement in an area where the flow gradients may actually be quite high.

A method that has proven to be an attractive alternative to either re-meshing or dynamic meshes is the transpiration boundary condition. Using the transpiration method, deflection and velocity boundary conditions are adjusted as necessary without modification to the existing CFD grid. Prior work has demonstrated that the transpiration method is an effective tool for incorporating a variety of structural deformations onto a CFD grid. Acknowledging the validity of the previously mentioned mesh-deformation techniques, the transpiration method involves only a minimal amount of computational overhead, which is of primary interest in an operational environment.

Further decrease in computational effort comes through the incorporation of a system-identification technique that greatly reduces the time required for the CFD-portion of the complete simulation⁵. Given an unsteady CFD solution subjected to a prescribed set of inputs, a system model can be constructed that essentially reduces the CFD portion of the code to a set of algebraic equations. These relatively simple equations can be solved on a time-scale that is several orders of magnitude shorter than the time required for an equivalent CFD solution.

The focus of this paper, therefore, is threefold. First, a quantitative investigation of the accuracy of the transpiration method is made using steady data from experimental results in the transonic regime. Second is a comparison between the computational flutter prediction and that obtained experimentally. Lastly, a complete aeroservoelastic model is constructed and used to demonstrate the use of a control surface to eliminate flutter.

Methodology

The primary research tools for the current effort are the STARS codes developed at NASA Dryden Flight Research Center⁶. The current version of STARS is the result of the evolution of the original STARS (STRUCTURAL ANALYSIS ROUTINES) computer code into an highly-integrated multidisciplinary tool for the analysis of a wide variety of 2D and 3D structures. This evolution involves the addition of several modules to the original STARS code. Each individual module, general by design, is integrated into an effective tool for the prediction of complicated aeroelastic and aeroservoelastic problems. These modules include: structures, heat transfer, linear aerodynamics, Euler-based CFD, controls engineering, and others.

An important component of this study is the implementation of the transpiration boundary condition into the CFD flow solver. Assuming that a surface normal has an x , y , and z component, a change in orientation is accomplished by changing the velocity boundary conditions on the affected nodes. This change in boundary condition comes in the form of an additional fluid velocity outside of the existing surface elements. This additional velocity affects the way the unsteady flow solver resolves the flow tangency boundary condition, see Figure 1 below.

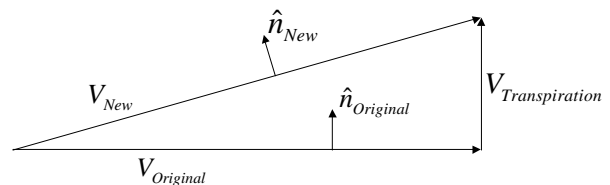


Figure 1: Transpiration Concept

In the above figure, $V_{Original}$ is the original tangential fluid velocity with normal \hat{n} . Through an aeroelastic or control surface deformation, for example, it is desired that the surface be deformed in such a way that it now has normal, \hat{n}_{New} . For the steady and unsteady cases, the flow tangency boundary condition is expressed in Equations (1) and (2), respectively.

$$V \cdot \hat{n} = 0 \quad (1)$$

$$V \cdot \hat{n} = V_b \cdot \hat{n} \quad (2)$$

Equation (1) simply states that the velocity normal to the body must be zero. Only slightly more complicated, Equation (2) states that the fluid velocity normal to the surface must be equal to the velocity of the body normal to itself. In other words, no flow can move through a solid surface. It is necessary to point out that the V_b mentioned here is not the same as $V_{Transpiration}$ shown in Figure 1.

Use of this transpiration boundary condition essentially solves the problems previously mentioned for re-meshing and dynamic meshes due to the following:

1. Since the same mesh is used throughout the entire computation, there is no computational penalty involved with re-defining and re-meshing the computational domain.
2. Since no physical mesh deformation takes place, the problems associated with mesh shearing disappear.
3. Only computational surfaces that undergo some sort of deflection must have their surface normals modified. This means that the vast majority of nodes within the computational domain are unmodified in the solution. This translates into substantial timesavings.

From above we see that the computational expense associated with the implementation of the transpiration boundary condition, compared with that of re-meshing or solving the spring-network necessary for dynamic meshes, is essentially negligible. Prior work has demonstrated that the use of the transpiration boundary condition is as accurate, within engineering accuracy, as an equivalent physical deformation.

The next key element in the complete aeroservoelastic simulation is the implementation of the system identification technique used to model the majority of the unsteady results presented in this paper. As mentioned previously, an unsteady CFD solution must be performed given a set of known inputs. An example of these prescribed velocities and resulting displacements are shown in Figure 2.

The corresponding generalized forces resulting from these modal inputs are unique to the system and can therefore be used to develop a model for the complete aeroservoelastic system. The model represents only the system's aerodynamics so it is independent of the structural stiffness, damping, control inputs, etc. It is also nondimensionalized such that, for a given Mach number, the dynamic pressure can be changed through a modification of the flow density, ρ . By successively changing this flow density, flutter points that would take many weeks of CPU time can now be found in less than one hour.

Each Mach number requires its own system model, but after the initial CFD solution is performed, the model can be used for aeroelastic as well as

aeroservoelastic applications. Further enhancing the usefulness of this model is the conversion of the system model into state-space representation. The state-space form of the equations can then be input into control analysis software, such as MATLAB®, and used to find eigenvalues, simulate the system dynamics, apply control surface deflections, etc.

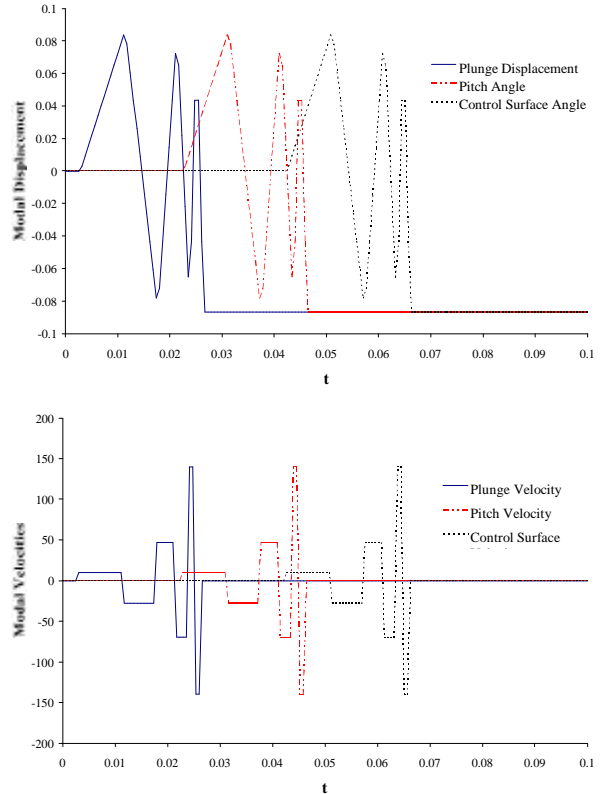


Figure 2: Displacement and Velocity Inputs for Unsteady CFD Flow Solver

Currently, it has proven convenient to use the model developed from the STARS CFD solution and convert it to state-space form. These state-space equations are subsequently input into MATLAB® where an appropriate control law is developed. This control law can then be implemented into the STARS codes for further validation using the time-marched Euler CFD solver.

Results

Results are presented for a benchmark configuration tested in the transonic regime. Beginning with steady flow about a wing with a deflected control surface, the transition is then made to the determination of the flutter boundary. From these steady and unsteady investigations, the progression is made to the full aeroservoelastic simulation where the movable control surface is incorporated into the unsteady CFD model as a means of flutter suppression.

Benchmark Model

The Structures Division of NASA Langley Research Center (LaRC) initiated the Benchmark Models Program (BMP) to obtain experimental data for the validation of unsteady CFD codes. A variety of models were tested in the NASA Langley Transonic Dynamics Tunnel (TDT)⁷. In the BMP program, two specific models are of interest. Each model has a rectangular planform with a NACA 0012 cross-section, 16-inch chord, and 32-inch span. The first model was simply a rigid rectangular wing fitted with pressure transducers over the surface of the wing. The second model is referred to as the BACT, standing for Benchmark Active Controls Technology. Though different models, each shares identical model dimensions, and instrumentation. The only practical difference between the two models is the presence of three control surfaces. These three control surfaces, two of which can be seen in Figure 3, are a trailing edge control surface, and upper and lower spoilers.

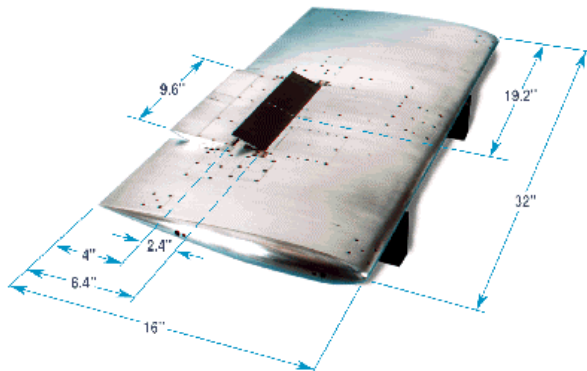


Figure 3: BACT Model Dimensions

The control surfaces are centered along the models 60% span (19.2 in), and has a length equal to 30% (9.6 in) of the wing's span. The trailing edge control surface has a width of 25% (4 in) model chord while the spoilers have a width of 15% (2.4 in) model chord.

The first model, the NACA 0012 wing, was tested in air and provided a large experimental database^{8,9,10}. This database included steady pressure measurements, unsteady pressure measurements during flutter, and flutter boundaries over a wide Mach number range. Tested in R-12, the BACT model's primary purpose was to provide additional data for the purposes of evaluating a CFD code's effectiveness in modeling the control surfaces illustrated above. For the scope of this paper, results are presented only for those experiments where the trailing edge control surface is used^{7,11}.

With the quality and amount of experimental data available, these models serve as the primary experimental benchmark to which all computational results obtained from the current research are compared.

Steady Comparisons

The starting point for the CFD solution is, of course, the definition of the computational geometry. Shown in Figure 4 is the unstructured finite-element mesh generated using STARS. The STARS CFD model has the same dimensions and proportions of the BMP models. Following from an earlier discussion, one notes only the inclusion of the trailing edge control surface. Grid density is increased in the region of the control surface to better account for the increased flow gradients as the control surface is deflected.

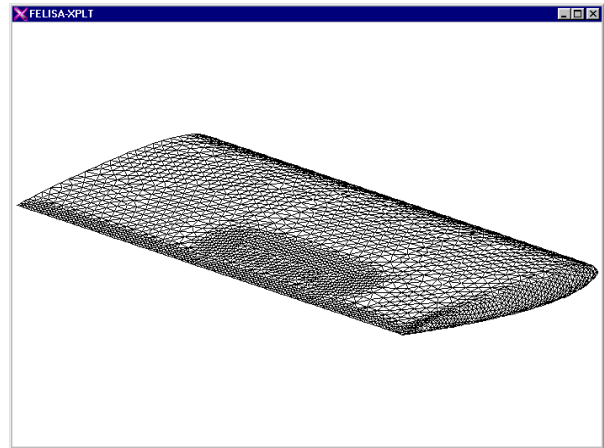


Figure 4: Surface Mesh Generated in STARS for the NACA 0012/BACT Wing

Steady comparisons for three different control surface deflection angles are given at Mach 0.77 and Mach 0.82. The Mach numbers and control surface deflections were chosen primarily on the basis of the availability of experimental data.

Mach 0.77 has the unique distinction of being the critical Mach number for this configuration. At this Mach number, flow accelerates from a free-stream value of Mach 0.77 to just sonic over the surface of the wing. Mach 0.82, however, has a well-defined transonic shock.

Comparisons with experimental data are made for 0° angle of attack with control surface deflection angles of 2°, 5°, and 10°. Note that a positive deflection angle causes an increase in lift, i.e. the control surface deflects downward. Quantitative comparisons come from observing the pressure distribution at the 60% span location, which corresponds the mid-span of the control surface.

Shown in Figure 5 are the comparisons between control surface deflections modeled using the transpiration boundary condition in STARS to actual experimental results obtained at Langley.

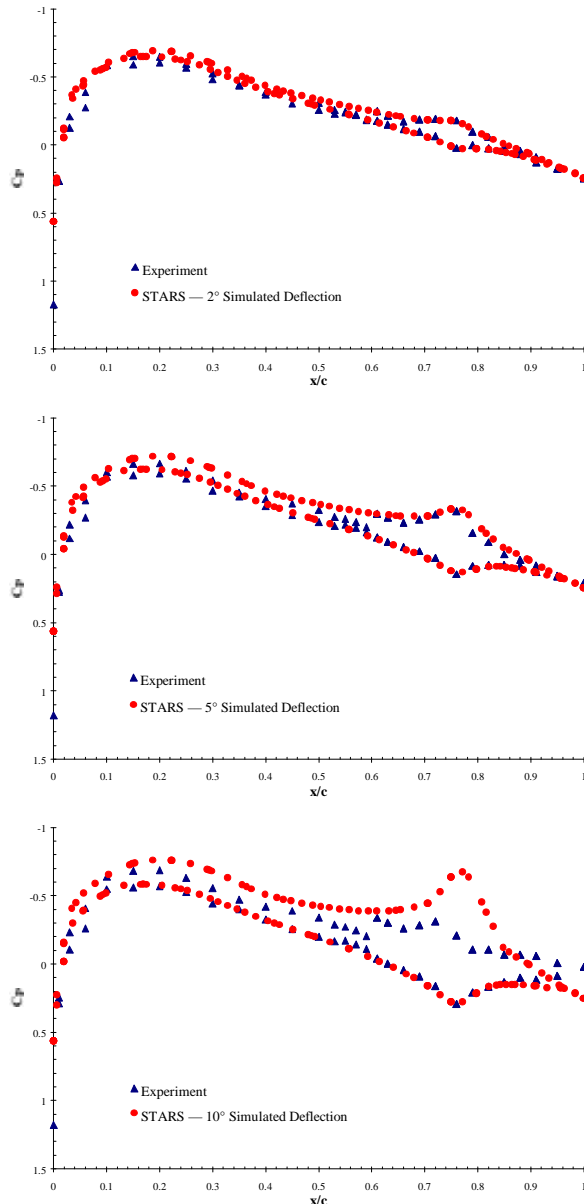


Figure 5: Comparison of Predicted and Experimental Chordwise Pressure Distributions at Mach 0.77, $\alpha = 0^\circ$, $\delta = 2^\circ, 5^\circ$, and 10°

For the 2° and 5° cases, one notes very good agreement between results obtained from STARS and experimental data. At $\delta = 10^\circ$ we observe the fact that significant viscous effects are present and the flow possibly separates as a result. This phenomenon, of course, cannot be modeled using the Euler-based CFD analysis in STARS.

We see in Figure 6, at Mach 0.82, very similar results to those given at Mach 0.77. Once again we note that for control surface deflections of 2° and 5° the good agreement between computational and experimental results.

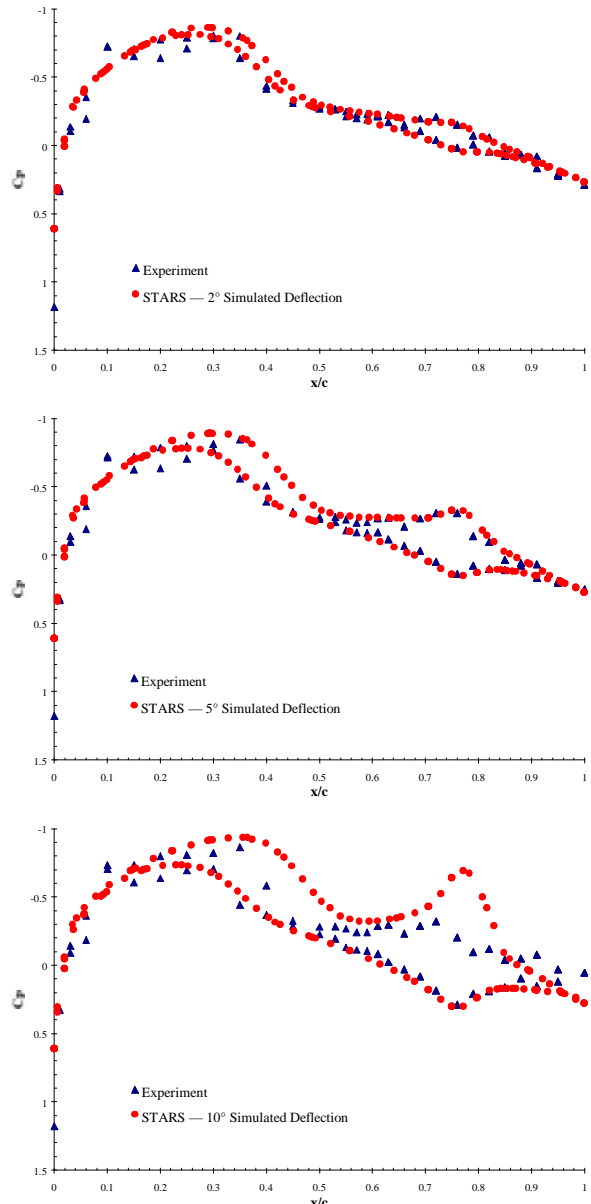


Figure 6: Comparison of Predicted and Experimental Chordwise Pressure Distributions at Mach 0.82, $\alpha = 0^\circ$, $\delta = 2^\circ, 5^\circ$, and 10°

Again we observe that the flow solver is unable to model the viscous effects obviously occurring with this relatively large 10° control surface deflection.

Slight differences can also be observed from slight deviations from nominal values in the experimental data. For example, the Mach number entered into STARS is exactly 0.77, experimentally, the Mach number ranged from 0.771 to 0.767. For a nominal Mach number of 0.82, experimental Mach numbers ranged from 0.81753 to 0.81933. Similarly for values of angle of attack and control surface deflection angle, values of α and δ were not exactly $0^\circ, 2^\circ$, etc. Any one

parameter should not make a significant difference, but a combination of each could build to an appreciable level of uncertainty. Again, with our computational model we are not subject to these little uncertainties, except for the case of machine round-off error. Also not modeled is a narrow transition strip mounted approximately 1" aft of the leading edge. Given the degree of uncertainty in the measurable experimental quantities, computational agreement with experiment remains satisfactory.

For practical applications, the 10° control surface deflection would generally be considered large in comparison with that necessary for flutter suppression. To better approximate experiment, however, the above results demonstrate a need for a deflection limit of no more than 5° during any ASE simulation.

Unsteady/Aeroelastic Comparisons

Before the aeroelastic and aeroservoelastic models are developed, one must again consider the experimental uncertainty present in the BACT model. As with any experimental measurement, we expect to see a certain amount of experimental uncertainty. In an effort to determine estimates for these uncertainties, communication with Mr. Robert C. Scott and Mr. Martin R. Waszak of the NASA Langley Research Center, provided valuable insight into the uncertainty of the measurement techniques^{11,12}.

Since the BACT wing is considered rigid, all of the stiffness terms arrive from the use of the pitch-and-plunge-apparatus (PAPA)¹³. The wing is reportedly mounted on the PAPA such that the elastic axis is coincident with the geometric center of the PAPA mount.

Experimental uncertainty in the determination of the center of gravity (CG) relative to the elastic axis (EA), however, proves to have a *very* significant effect on the prediction of the flutter boundary. In the literature, the CG's location relative to the elastic axis was reported, at best, to be nearly coincident with the mid-chord of the wing. The reported value of the inertial coupling between the pitch and plunge modes ($S_{h,a}$) is 0.0142 slug-ft¹⁴. Using Equation (3) below, we can estimate the relative location of the CG to the EA.

$$S_a = m \cdot x_{cg} \quad (3)$$

Using the value of S_a that was reported by Waszak and a mass of 6.08 slugs, we calculate the distance from the EA to the CG to be 0.028 inches.

From available data, it is possible to construct a simple model from which we could quickly evaluate the importance of some of the above parameters. Using a simplified aerodynamic model and solving the equations of motion using the p -method, we can quickly solve for the divergence speed. Since the flutter speed can be solved for explicitly in the p -method, parametric studies can be done very quickly.

Shown below in Figure 7 we see the effect of three different parameters on the flutter prediction.

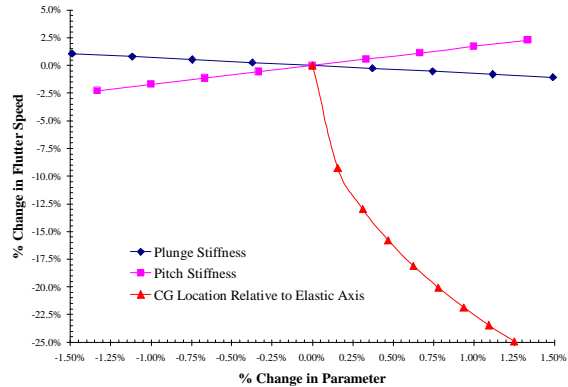


Figure 7: Effect of K_h , K_a , and x_{cg} on Flutter Prediction

In the above figure, the horizontal axis represents small deviations from nominal values for plunge and pitch stiffness, K_h and K_a , and x_{cg} , which is a measure of the distance from the elastic axis to the center of gravity, measured positive aft. As is shown, small changes in both plunge and pitch stiffness effect little change in the flutter prediction, $\sim \pm 2.5\%$. Small changes in x_{cg} , however, influence the flutter prediction significantly. Using the above model, changes in x_{cg} on the order of 1% can change the flutter prediction by over 20%.

Uncertainty in flutter prediction is magnified by the fact that the center of gravity is approximately coincident with the elastic axis. For comparison with experimental data, small variations in each of these parameters can add up to large differences in flutter prediction. This is a point to be considered as the aeroelastic development continues.

As mentioned earlier, the use of the system identification technique significantly decreases the runtime required to determine the flutter boundary. Recall that the first step in this process is to obtain an unsteady CFD solution for the variable multi-step sequence. For the modal inputs shown in Figure 2, we see the actual response shown in Figure 8. In this figure we see two generalized forces (lift and pitch moment) plotted against time. The continuous line represents the Euler solution from STARS. The individual points represent the solution obtained from the discrete-time model resulting from the system identification technique. This particular case is at Mach 0.82, but similar results exist for each Mach number investigated.

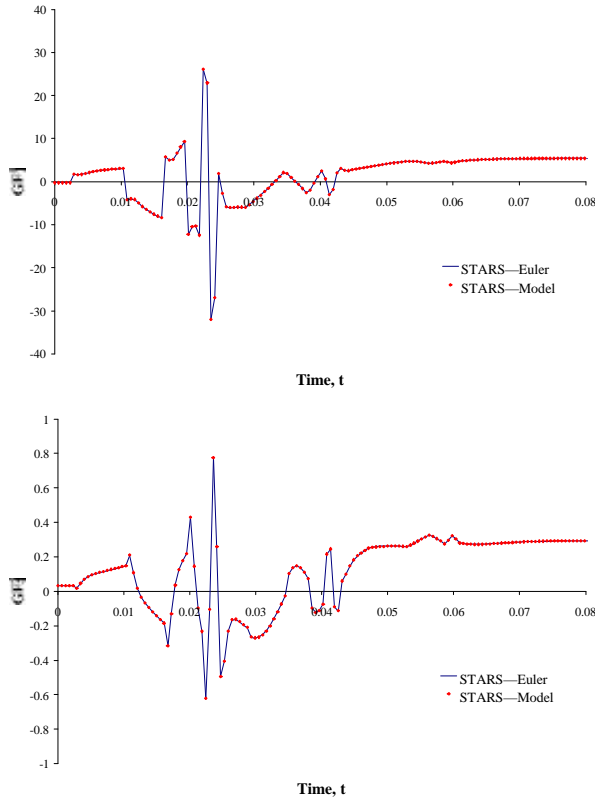


Figure 8: Model Training Data at Mach 0.82, $q=159.9$ psf

Having the training data and models for Mach numbers ranging from 0.51 to 0.82, an estimate of the flutter boundary is now possible. Recall, however, the sensitivity of the flutter point suggested by the proximity of the CG and EA. Shown in Figure 9 are three such boundaries as predicted by STARS using the system model and three different values of x_{cg} : 0.028", 0.08", and 0.1".

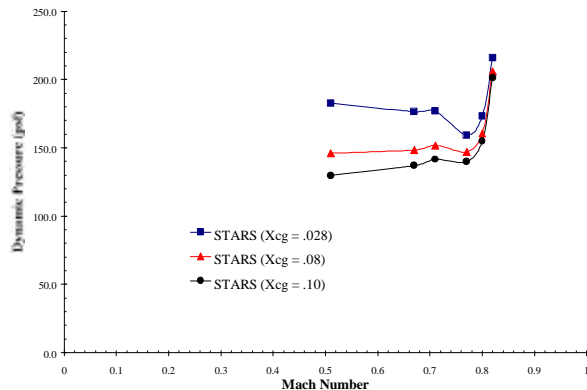


Figure 9: STARS Flutter Boundary Predictions for Different x_{cg} Locations

A value of 0.028", recall, comes from Equation (3). Experimental uncertainty in the determination of x_{cg} was estimated to be no more than 0.1", hence the third

value of x_{cg} shown¹². The value of 0.08", as is shown later, was the best fit to the experimental data. As expected, the further aft the CG is relative to the EA, the lower the flutter prediction

The differences noted are not too dissimilar from the simple p -method analysis shown in Figure 7. In fact, differences range from as high as 29% at Mach 0.51 to 7% at Mach 0.82. The interesting trend here is the fact that this sensitivity to x_{cg} tends to decrease with increasing Mach number.

Shown in Figure 10 is a comparison of the flutter boundary predictions of both the NACA 0012 and BACT wings and the STARS model with an x_{cg} of 0.08".

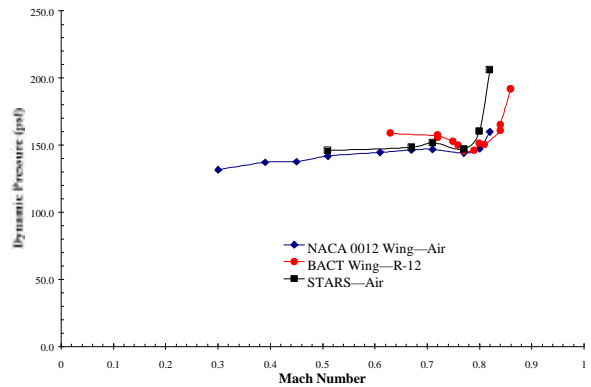


Figure 10: STARS Flutter Prediction Compared with Experimental Results from the NACA 0012 (air) and BACT (Freon) Wings

From Mach 0.51 to 0.77, predicted values of $q_{flutter}$ were less than 4% different than experiment. As was noticed in the experimental data, the predicted flutter boundary increased sharply past the transonic dip. Though slightly higher than observed through experiment, predicted results compared reasonably well.

Aeroservoelastic Analysis

The natural extension of work done to this point is to use the control surface on the BACT wing to suppress flutter. A description of the development and implementation of the control law into STARS is given in the next few paragraphs.

Before considering the development of the control system itself, it is first useful to analyze the frequency response of the ASE model to further see how well the model represents the physical system. The frequency response, shown in Figure 11, results from a spectral analysis with the output being the trailing edge acceleration and the input being control surface deflection. This figure compares well with data obtained from other computational models and experimental data¹⁴.

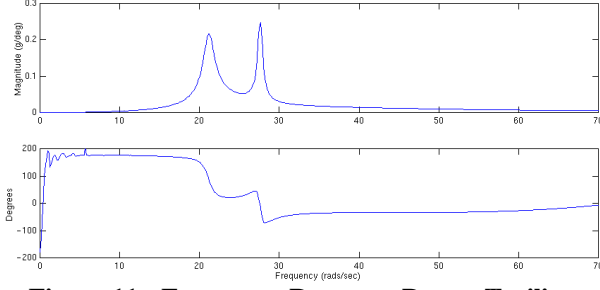


Figure 11: Frequency Response Due to Trailing Edge Control ($q=125$ psf)

In general, any control law will have as its output a desired flap position. Knowing that the control surface itself has mass and inertia, one cannot expect that the control surface would instantaneously respond to a specified control input. The need, therefore, for an actuator model is evident. Waszak reports the control surface actuator's transfer function as (4) where k (1.02 deg/deg) is the actuator gain, z (.56) is the damping ratio, w (rad/sec) is the 2nd order natural frequency, \mathbf{d} is the desired control surface deflection, and \mathbf{d} is the actual resulting deflection¹⁵.

$$\frac{\mathbf{d}}{\mathbf{d}_s} = \frac{k\mathbf{w}^2}{s^2 + 2\mathbf{W}s + \mathbf{w}^2} \quad (4)$$

For our purpose, however, it is more convenient to move from the frequency domain back to the time domain. The corresponding differential equation is shown in (5).

$$\ddot{\mathbf{d}} + 2\mathbf{W}\dot{\mathbf{d}} + \mathbf{w}^2\mathbf{d} = k\mathbf{w}^2\mathbf{d}_s \quad (5)$$

To begin putting the above equation into state-space format, we'll make the following substitutions:

$$x_1 = \mathbf{d} \quad \text{and} \quad x_2 = \dot{\mathbf{d}} = \dot{x}_1.$$

Taking derivatives of these equations results in the following: $\dot{x}_1 = \dot{\mathbf{d}} = x_2$

and $\dot{x}_2 = \ddot{\mathbf{d}} = \ddot{x}_1$. Using these relationships, we rewrite (5) in the following form:

$\dot{x}_2 + 2\mathbf{W}x_2 + \mathbf{w}^2x_1 = k\mathbf{w}^2\mathbf{d}_s$. We now have two first-order differential equations which we can write in state-space format, see (6).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\mathbf{w}^2 & -2\mathbf{W} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ k\mathbf{w}^2\mathbf{d}_s \end{bmatrix} \quad (6)$$

To actually implement these equations into STARS, the time-derivatives are replaced by the following relationships, shown in (7) and (8), where n is the current value, and $n-1$ represents past values.

$$\frac{x_1^n - x_1^{n-1}}{\Delta t} = x_2^{n-1} \quad (7)$$

$$\frac{x_2^n - x_2^{n-1}}{\Delta t} = k\mathbf{w}^2\mathbf{d}_s - 2\mathbf{W}x_2^{n-1} - \mathbf{w}^2x_1^{n-1} \quad (8)$$

Solving each for the current values of x_1 and x_2 yields (9) and (10).

$$x_1^n = x_1^{n-1} + x_2^{n-1}\Delta t \quad (9)$$

$$x_2^n = (k\mathbf{w}^2\mathbf{d}_s - 2\mathbf{W}x_2^{n-1} - \mathbf{w}^2x_1^{n-1})\Delta t + x_2^{n-1} \quad (10)$$

Up to this point, the desired control surface position has been arbitrary. For our purpose, the desired flap angle will be a function of plunge displacement and velocity, and pitch displacement and velocity. The resulting control law is shown in (11), where the gains K_i are not necessarily absolute.

$$\mathbf{d}_s = K_1h + K_2\dot{h} + K_3q + K_4\dot{q} \quad (11)$$

Given the range of Mach numbers, and simply the nature of the BACT system, it makes sense that a single controller would be insufficient for each particular case. In general, one would need to consider both the Mach number and dynamic pressure when designing a particular controller. For the purpose of this paper, demonstrating ASE analysis within STARS is the primary goal.

MATLAB[®] was used in the controls analysis. Recall that we can construct a discrete-time representation of both the structural dynamics and aerodynamics from the system model incorporated in STARS. The coupling of these two systems, of course, represents the aeroelastic system. The addition of control, through the trailing edge control surface on the BACT wing, allows the relatively quick development of an ASE model capable of suppressing flutter. Shown in Figure 12 is the block-diagram representation of the complete BACT ASE model used in MATLAB[®], SIMULINK[®].

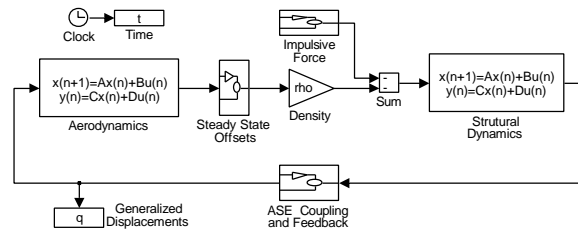


Figure 12: Block Diagram of BACT ASE Model

Recall that the discrete-time model is valid for only a single Mach number, but the dynamic pressure may be changed via a change in the density, ρ .

Equations (4)-(11) show how the actuator model was taken from the Laplace domain, into a form that could be incorporated into STARS. Similarly for the MATLAB[®] model the continuous-time form of the actuator model, (4) and (12), must be represented in a discrete-time format. Although not completely necessary, having the discrete time actuator model will help reduce the deviation between the STARS and MATLAB[®] models.

Beginning with the continuous-time actuator model shown in (12), we use a trapezoidal approximation when converting to discrete-time.

$$G(s) = \frac{k\mathbf{w}^2}{s^2 + 2\mathbf{W}s + \mathbf{w}^2} \quad (12)$$

This particular conversion requires that the Laplace variable, s , be replaced by (13), where T is the discrete time-step.

$$s = \frac{2(z-1)}{T(z+1)} \quad (13)$$

Performing the substitution and simplifying, we get the following discrete-time transfer function which can be entered directly into MATLAB®.

$$G(z) = \frac{k(\mathbf{w}T)^2(z^2 + 2z + 1)}{(4\mathbf{W}T + (\mathbf{w}T)^2 + 4)z^2 + (2(\mathbf{w}T)^2 - 8)z + (\mathbf{w}T)^2 - 4\mathbf{W}T + 4}$$

Having the MATLAB® model and an equivalent model integrated into the STARS codes, comparisons between the two are now possible. Figure 13 shows the open-loop response of the BACT wing at Mach 0.77, $q=155.4$ psf, where flutter occurs at approximately 147 psf.

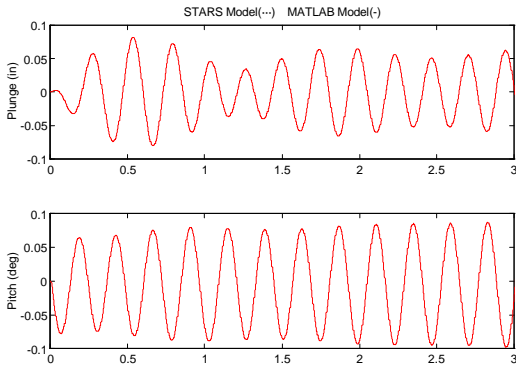


Figure 13: Open Loop Response Comparison Between STARS and MATLAB® Models (Mach 0.77, $q=155.4$ psf)

Implementing the following control law in each of the models gives the following closed-loop response for the same dynamic pressure and initial conditions:

$$\mathbf{d}_s = -5\mathbf{h} - 2\mathbf{a} - \dot{\mathbf{a}}$$

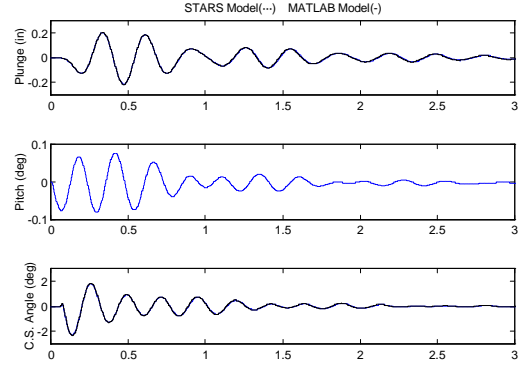


Figure 14: Closed Loop Flutter Suppression Comparison Between STARS and MATLAB® Models (Mach 0.77, $q=155.4$ psf)

Similarly, for the Mach 0.82 case, Figure 15 shows good agreement between the STARS and MATLAB® models for the open-loop case. In this particular case, the Mach number is 0.82, $q=215.3$ psf., where STARS predicts $q_f=206.2$ psf. These open-loop time histories give a qualitative feel for the extent of the instability at this particular dynamic pressure.

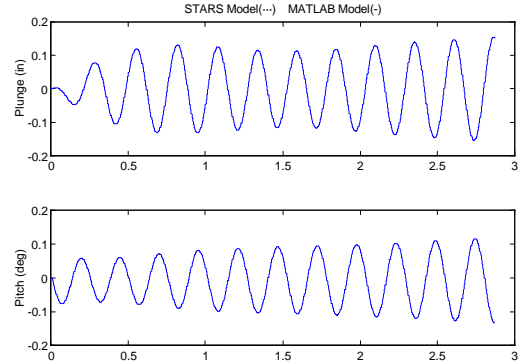


Figure 15: Open Loop Response Comparison Between STARS and MATLAB® Models (Mach 0.82 $q=215.3$ psf)

For the Mach 0.82 case, control was obtained using only pitch damping. Hence the control law for this particular case was simply $\mathbf{d}_s = \dot{\mathbf{a}}$. Time histories from the STARS and MATLAB® models are given below in Figure 16.

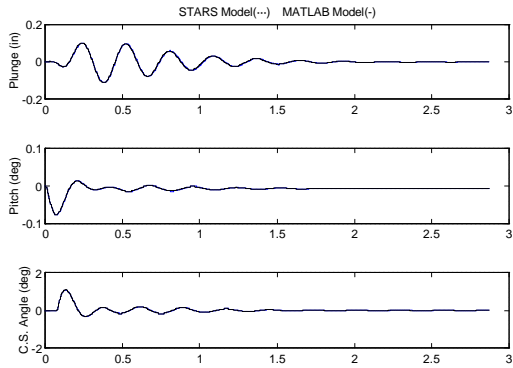


Figure 16: Closed Loop Flutter Suppression Comparison Between STARS and MATLAB® Models (Mach 0.82, $q=155.4$ psf)

What has been demonstrated up to this point is the fact that the STARS and MATLAB® models agree well with one another. Recall that each model is based training data obtained from STARS. The true measure of the models validity is, of course, the extent to which it accurately represents the results obtained from the CFD flow solver.

Validation is shown for only a single Mach number, 0.82, and dynamic pressure, 215.3 psf. For each of the model time-histories shown above, the three-second trace was obtained from in less than a minute. The abbreviated, 1.5 second, time-history, shown in Figure 17, took over two weeks on a DEC-Alpha 500 MHz machine running Windows NT 4.0. Given these statistics, one quickly appreciates the substantial timesavings offered by the model.

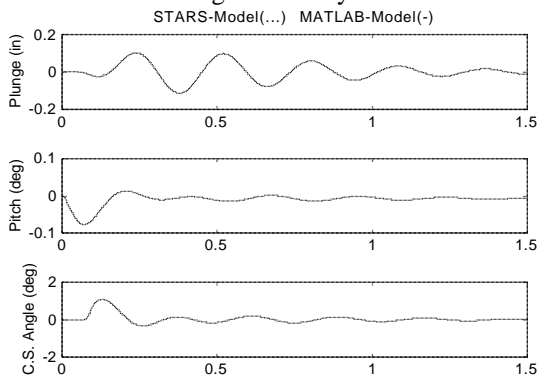


Figure 17: Euler Validation of Closed Loop Flutter Suppression (Mach 0.82)

The above time-history is the result of approximately 2600 CFD steps. Across the entire plot, agreement between the computational model and the STARS model is very good. Given the number of steps, the potential for accrued error as the analysis progresses is substantial. Good agreement, therefore, further enhances confidence in the model.

Concluding Remarks

The goal of this study was to implement an ASE simulation into STARS and compare computational results with experimental results from a benchmark configuration. Within STARS, the transpiration method is applied to account for structural deformations. Additionally, through a system identification process, the coupled ASE system was modeled as a discrete-time dynamic system. From this model, the time required for aeroelastic, and aeroservoelastic analyses was significantly reduced.

Using the above techniques, computational results were compared with the benchmark test case, the BACT wing from NASA Langley. Results included steady flows, which demonstrated the effectiveness of the transpiration method when applied to control surface deflections. The aeroelastic analysis demonstrated good agreement with experiment between flutter predictions across a wide Mach number range. Finally, the ASE analysis was successfully implemented and validated within the STARS computer codes.

Results in this study were obtained within the transonic flow regime. Certain difficulties arise and complicate the modeling procedure when considering this range of Mach numbers. For the BACT wing at Mach numbers greater than 0.77 transonic shocks are present on the upper and lower surfaces. The model accounts for the existence of this statically non-linear phenomenon. Dynamic linearity is obtained in the coupled ASE simulation, where the shocks move due to pitch and plunge motions as well as control surface deflections.

Control surface movement was limited by the desire to stay within the limits of the inviscid code's ability to correctly model the deflection. This limitation also assured that the deviations from the nominal case were kept small.

Overall, the implementation of the ASE model into STARS provided good agreement with experimental data for each area investigated. While results compared well for one specific case, the procedure is general enough to allow successful application to different geometries and Mach numbers.

Acknowledgements

Funds for the support of this study have been allocated through the NASA-Ames University Consortium Office, under Interchange Number NCC2-5105, and Oklahoma State University.

References

- ¹Lee-Rausch, E.M., Batina, J.T., "Wing Flutter Boundary Prediction Using Unsteady Euler Aerodynamic Method," *Journal of Aircraft*, Vol. 32, No. 2, March-April 1995, pp. 416-422.

- ²Fisher, C.F., and Arena, A.S., “On the Transpiration Method for Efficient Aeroelastic Analysis Using an Euler Solver,” AIAA Paper 96-3436.
- ³Stephens, C.H., Arena, Jr., A.S and Gupta, K.K., “Application of the Transpiration Method for Aeroservoelastic Prediction Using CFD,” AIAA Paper 98-2071.
- ⁴Stephens, C.H., “CFD-Based Aeroservoelastic Predictions on a Benchmark Configuration using the Transpiration Method”, *Masters Thesis*, Oklahoma State University, July 1998.
- ⁵Cowan, T.J., “Efficient Aeroelastic CFD Predictions Using System Identification”, *Masters Thesis*, Oklahoma State University, May 1998.
- ⁶Gupta, K.K., “STARS – An Integrated General-Purpose Finite Element Structural, Aeroelastic, and Aeroservoelastic Analysis Computer Program”, NASA TM-4795, May 1997.
- ⁷Scott, R.C., Hoadley, S.T. Wiesman, C.D. and Durham, M.H., “The Benchmark Active Controls Technology Model Aerodynamic Data,” AIAA Paper 97-0829.
- ⁸Rivera, A.J., Jr., Dansberry, B.E., Farmer, M.G., Eckstrom, C.V., Seidel, D. A., and Bennett, R.M., “Experimental Flutter Boundaries With Unsteady Pressure Distributions for the NACA 0012 Benchmark Model,” NASA-TM-104072, 1991.
- ⁹Rivera, A.J., Jr., Dansberry, B.E., Bennett, R.M, Durham, M.H., and Silva, W.A., “NACA 0012 Benchmark Model Experimental Flutter Results With Unsteady Pressure Distributions,” NASA-TM-107581, 1992.
- ¹⁰Rivera, A.J., Jr., Dansberry, B.E., Durham, M.H., Bennett, R.M., and Silva, W.A., “Pressure Measurements on a Rectangular Wing With a NACA 0012 Airfoil During Conventional Flutter,” NASA-TM-104211, 1992.
- ¹¹Scott, R.C., Aerospace Engineer, NASA Langley Research Center. Personal Communications, 1998.
- ¹²Waszak, M.R., Aerospace Research Engineer, NASA Langley Research Center. Personal Communications, 1998.
- ¹³Farmer, M.G., “A Two-Degree-of-Freedom Flutter Mount System with Low Damping for Testing Rigid Wings at Different Angles of Attack,” NASA TM-83302, 1982.
- ¹⁴Waszak, M.R., “Modeling of the Benchmark Active Controls Technology Wind-Tunnel Model for Application to Flutter Suppression,” AIAA Paper 96-3437.
- ¹⁵Waszak, M.R. and Fung, J., “Parameter Identification and Analysis of Actuators for the BACT Wind-Tunnel Model,” AIAA Paper 96-3362.