Active control laws for computational aeroelasticity

Al. Dumitrache
Institute of Mathematical Statistics and Applied Mathematics
Calea 13 Septembrie no. 13, sect. 5, Bucharest, 050711, Romania

Abstract
The major objective of this paper is to describe a general methodology to design control laws in the context of a computational aeroelasticity environment. The technical approach involves employing a systems identification technique to develop an explicit state-space model for control law design from the output of a computational aeroelasticity code. Although there are many control law design techniques available, the standard Linear Quadratic Guassian (LQG) technique is employed in this paper. The computational aeroelasticity code is modified to accept control laws and perform closed-loop simulations. Numerical results for flutter suppression of the Benchmark Active Control Technology wind-tunnel model are given to illustrate the approach.

1. Introduction

Aeroelasticity has been and continues to be an extremely important consideration in many aircraft designs. The control of aeroelastic response through feedback to control surfaces or more recently through feedback to active materials, is an alternative to "passive control" through increased stiffness. Currently, most aeroelasticity considerations are routinely addressed using linear aeroelastic (i.e. linear aerodynamic and linear elasticity) models. However, within the last few decades, a significant increase in advancing methods to consider nonlinear aeroelasticity, especially nonlinear aerodynamics in the transonic region, has taken place.
Computational aeroservoelasticity involves coupling structural dynamics, computational fluid dynamics, and active control systems together. Batina and Yang[1] were perhaps the first researchers to examine control of an aeroelastic system in a computational aeroelasticity environment for transonic flow. They conducted studies with a 2-d airfoil with plunge and pitch degrees-of-freedom and a 2-d small-disturbance transonic CFD code. The effect of a simple constant gain control law utilizing displacement, velocity, and acceleration feedback on the time responses was determined. Comparison with linear theory indicated that the frequency and damping values were significantly different for transonic and linear subsonic theory results. References [2-9,17] are other examples of research in control of aeroelastic systems within a computational aeroelasticity environment. Similar to Batina and Yang, these studies also only involve varying the gains of simple feedback control laws to study their effect on the response of the aeroelastic system. Two-dimensional and three-dimensional small-disturbance and Euler CFD codes are used in these studies. The studies show that feedback control can be effective in suppressing transonic flutter.
Guillot and Friedman[11,12] employed adaptive control theory to design control laws using a CFD technique. A 2-dimensional airfoil model, with a trailing-edge control surface, is used with an Euler CFD code to perform the computational aeroelastic solutions. An adaptive control law was used because of the assumed nonlinear behavior of the system in the presence of nonlinear transonic flow with large shock motions. The adaptive control law involves identifying a linear auto-regressive moving average (ARMA) model and then determining an optimal full-state control law. An adaptive control law was shown to be quite effective in suppressing transonic flutter with strong shocks.

2. Description of technical approach

The overall methodology begins with performing a computational aeroelasticity simulation (uncontrolled) with prescribed control surface inputs to obtain a set of corresponding output time histories. The next step is to employ a system identification technique, using the time histories of outputs and inputs from the first step, to determine an "equivalent linear system" for use as a control law design model. Next, design of a control law design can be performed using any control law design technique. Finally, the control law is evaluated in the computational aeroelasticity simulation. If the control law performance is not adequate, the control law can be redesigned and evaluated again until the desired performance is obtained.

2.2 Computational Aeroelasticity Simulation

Computational aeroelasticity simulation involves integrating the structural, aerodynamic (CFD), and for this research, control equations simultaneously. This paper focuses on the transonic case where the aerodynamic fluid flow equations contain nonlinear terms.
The particular code that is employed is the CAP-TSD[13,14] code that has been developed at the NASA Langley Research Center. The CAP-TSD code is a finite-difference code which solves the transonic small-disturbance equation. The primary outputs of the CAP-TSD code are time histories of the pressures and the generalized coordinate displacements, velocities, and accelerations. It has been used on a wide variety of configurations for steady and unsteady pressure distribution calculations and for calculating transonic flutter characteristics, including nonlinear limit-cycle instabilities.
The basic equations of motion implemented in CAP-TSD are:

$$[M]\{\ddot{q}\}+[C]\{\dot{q}\}+[K]\{q\}=\{F\}\,. \tag{1}$$

The primary difference between state-of-the-art linear aeroelasticity methods and computational aeroelasticity is in the computation of the aerodynamic pressure ΔC_p that is used in computing the generalized aerodynamic force vector $\{F\}$.
The CAP-TSD code is a finite difference program that solves the general-frequency modified TSD potential equation

$$M_\infty^2(\phi_t + 2\phi_x)_t = \left[(1-M_\infty^2)\phi_x + F\phi_x^2 + G\phi_y^2\right]_x + \left(\phi_y + H\phi_x\phi_y\right)_y + \left(\phi_z\right)_z \tag{2}$$

where M_∞ is the freestream Mach number, ϕ is the disturbance velocity potential, and the subscripts of ϕ represent partial derivatives.
Several choices are available for the coefficients *F*, *G*, and *H*, depending upon the assumptions used in deriving the TSD equation. In this paper, the coefficients are defined as

$$F=-\frac{1}{2}(\gamma+1)M_\infty^2;\;\; G=\frac{1}{2}(\gamma-3)M_\infty^2;\;\; H=-\frac{1}{2}(\gamma-1)M_\infty^2,$$

where γ is the ratio of specific heats of the aerodynamic fluid. The linear potential equation can be solved by simply setting *F*, *G*, and *H* equal to zero.
Equation 2 is solved within CAP-TSD by a time-accurate approximate factorization (AF) algorithm developed by Batina[13]. The algorithm consists of a Newton linearization procedure coupled with an internal iteration technique. The CAP-TSD code is capable of treating configurations with multiple lifting surfaces and bodies. A relatively simple Cartesian grid is input along with the coordinates defining the geometry of the configuration and the corresponding surface slopes. After the potential is calculated at each time step, the pressure coefficient is calculated by $C_p = -2\phi_x - 2/U_\infty\phi_t$. The pressure coefficient is employed to calculate the generalized force vector at each time step.
Equation 1 can be rewritten in state-space form as

$$\dot{X} = [A]X + [B]u\,, \tag{3}$$

where

$$X=(q,\dot{q})^T,\;\; [A]=\begin{bmatrix}0 & I\\ -M^{-1}K & -M^{-1}C\end{bmatrix},\;\; [B]=\begin{bmatrix}0\\ M^{-1}\end{bmatrix} \text{ and } u=F\,,$$

The numerical algorithm (Edwards , Bennet[5]) employed in CAP-TSD, for solving equation 3 is

$$X_{n+1} = \phi X_n + \theta B(3u_n - u_{n-1})/2$$

where $\phi = e^{A\Delta\tau}$, $\theta = \int_0^{\Delta\tau} e^{A(\Delta T-\tau)}d\tau$ and ΔT is the integration time step.

2.3 Control Law Design Model Development

Most control law design methods require an explicit mathematical model of the system to be controlled. A CASM provides time histories of the variables of an aeroelastic system, but does not generate an explicit mathematical model of the system. CASM are analogous to performing experimental investigations where the only direct outputs are time responses. Therefore, to employ the various control law design methods that are available, a control design mathematical model of a "computational aeroelasticity system" (CAS) must be developed.
System identification techniques[15,16] are widely employed for developing a mathematical model given experimental data.
Therefore, since a CAS simulation is analogous to performing an experiment, system identification is a logical choice. The Observer/Kalman Filter Identification (OKID) technique[18] was developed primarily for development of a mathematical model for control law design. Because the OKID technique was developed primarily for identifying models for control law design, it is the technique employed in this paper.
One of the keys to the OKID algorithm is the introduction of an observer into the identification process. The fist step of the process is the calculation of the observer Markov parameters. Then the system Markov parameters are obtained.
Consider a discrete time state-space model of a system described by a set of first order difference equations of the form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k . \end{aligned} \tag{4}$$

Note that the triplet $[A,B,C]$ is not unique, but can be transformed through any similarity transform to another set of coordinates. Solving for the output y_k in terms of the previous inputs, with the assumption that the system is initially at rest, i.e. $x_0 = 0$, yields

$$y_k = \sum_{i=0}^{k} h_i u_{k-i} , \tag{5}$$

where the parameters $h_0 = D,\ h_1 = CB,\ h_2 = CAB, ..., h_k = CA^{k-1}B,\ k = 1,2,3...$ are the system Markov parameters, which are also the system pulse response samples.
To reduce the number of Markov parameters needed to adequately model a system, an observer is introduced into the OKID technique. Adding and subtracting the term $\overline{G}\, y_k$ to the right hand side of equation (4) yields

$$\begin{aligned} x_{k+1} &= \left(A+\overline{G}C\right)x_k + \left(B+\overline{G}D\right)u_k - \overline{G}y_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{6}$$

. The matrix $\overline{G}$ can be interpreted as an observer gain. The parameters defined as

$$\overline{Y}(k) = C\left(A+\overline{G}C\right)^{k-1}\left[B+\overline{G}D, -\overline{G}\right] = [\beta_k,\ \alpha_k] \tag{7}$$

are the Markov parameters of an observer system. Consider the special case where $\overline{G}$ is a deadbeat observer gain such that all eigenvalues of $A+\overline{G}C$ are zero, the observer Markov parameters will become identically zero after a finite number of terms. For lightly damped systems, this means that the system can be described by a reduced number of observer Markov parameters. Furthermore, an unstable system can be represented using this technique. This is obviously a major advantage for this research.
The Markov parameters are solved using a least squares technique. The observer state equations (6) can be rewritten as

$$\begin{aligned} x_{k+1} &= \hat{A}x_k + \hat{B}v_k \\ y_k &= Cx_k + Du_k \end{aligned} \qquad \textbf{(8)}$$

where $\hat{A} \equiv A + \overline{G}C$, $\hat{B} \equiv \left[B + \overline{G}D, -\overline{G}\right]$ and $v_k = \begin{bmatrix} u_k \\ y_k \end{bmatrix}$.

Similar to (5), but in matrix form, a data set of $N+1$ sampled outputs can be represented as $\overline{y} = \overline{Y}\overline{V}$ where

$$\overline{y} = [y_0\ y_1 \dots y_N],\ \overline{Y} = [\beta_0, \beta_1, \alpha_1, \beta_2, \alpha_2, \dots \beta_p, \alpha_p\ ]$$

$$\overline{V} = \begin{bmatrix} u_0 & u_1 & \dots & u_p & u_{p+1} & \dots & u_N \\ & v_0 & \dots & v_{p-1} & v_p & \dots & v_{N-1} \\ & & . & . & . & . & . \\ & & & v_0 & v_1 & \dots & v_{N-p} \end{bmatrix}$$

The observer Markov parameters can be solved by $\overline{y} = \overline{Y}\overline{V}^+$ where $\overline{V}^+$ is the pseudo-inverse of $\overline{V}$. The actual system Markov parameters are determined from the observer Markov parameters using a recursive formula. By partitioning $\overline{Y}$ and using the definition of the system Markov parameters, (equation 7), the system Markov parameters Y are recovered by

$$Y_0 = \overline{Y}_0 = D,\ \ Y_K = \beta_k - \sum_{i=1}^{k} \alpha_i Y_{k-i} \ \text{ for } k = 1,2,\dots,p$$

$$Y_K = -\sum_{i=1}^{k} \alpha_i Y_{k-i} \ \text{ for } k = p+1,\dots,\infty$$

Then a state-space model of a system is developed, employing the system Markov parameters, using the Eigensystem Realization Algorithm (ERA) , (Juang, Pappa)[19]. The ERA begins with determining the singular value decomposition of a Hankel matrix with entries that are the Markov parameters.

$$H(k-1) = \begin{bmatrix} Y_k & Y_{k+1} & \dots & Y_{k+\beta-1} \\ Y_{k+1} & Y_{k+2} & \dots & Y_{k+\beta} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ Y_{k+\alpha-1} & Y_{k+\alpha} & \dots & Y_{k+\alpha+\beta-2} \end{bmatrix}; \quad H(k-1) = \begin{bmatrix} C \\ CA \\ . \\ . \\ . \\ CA^{\alpha-1} \end{bmatrix} A^{k-1}\left[B \quad AB \quad A^2B \quad \dots \quad A^{\beta-1}B\right].$$

where α and β are integers. The Hankel matrix can be decomposed into the Observability matrix, a state transition matrix, and the Controllability matrix. This truncated Hankel matrix is then used to reconstruct the triplet $[A, B, C]$. The order of the system is determined by the singular value decomposition of $H(0)$. $H(0) = U\,\Xi V^T$ where the columns of U and V are orthonormal, Ξ. is an $n \times n$ diagonal matrix of positive singular values, and n is the order of the system.
A discrete-time minimal-order realization of the system is:

$$A = \Sigma^{-1/2} U^T H(1) V \Xi^{-1/2}, \quad B = \text{first}\, m \text{ columns of } \Xi^{1/2} V^T, \quad C = \text{first } r \text{ rows of } U\Xi^{1/2}$$

where m is the number of inputs and r is the number of outputs.

2.4 Control law design

There are many control law design methods available. These range from classical control law design to the LQG method to H_{∞}. robust control law design methods to nonlinear control law design methods. Because of its ease of use, the LQG design method is employed in this paper. Basically the LQG design process involves minimizing a cost function of the form

$$J = \frac{1}{2}\int_0^{\infty}\left[y^T Q y + u^T R u\right]dt \qquad \textbf{(9)}$$

where *y* is an output vector (e.g. accelerations, loads) of the system and *u* is the control input. The resulting control law is $u = -G\,x$ where G is the state feedback gain matrix that minimizes the cost function and x is the state vector. Since the states of a system are generally not all available for feedback, a Kalman filter is employed to estimate the states. The resulting control law is of the form:

$$\begin{aligned} \{\dot{\hat{x}}\} &= [\hat{A}]\{\hat{x}\} + [L]\{y\} \\ \{u\} &= [-G]\{\hat{x}\} \end{aligned} \qquad \textbf{(10)}$$

where $\{\hat{x}\}$ is the estimate of the state vector x and L is the Kalman filter gain matrix.

3. Results and discussion

The example employed in this paper to demonstrate the overall control law design methodology is active flutter suppression for the Benchmark Active Controls Technology (BACT)[20] wind-tunnel model. Selecting this model has the advantages of available analytical and experimental data and control law design results for linear models. The BACT model is a rigid, rectangular wing with a NACA 0012 airfoil section. The rectangular wing has a span of 0.812 m and a chord of 0.406 m and therefore an aspect ratio of 2. It is equipped with a trailing-edge control surface and upper and lower surface spoilers that are controlled independently by hydraulic actuators. Only the trailing-edge control surface is employed in this paper.

For the control law designs, an accelerometer located near the outboard trailing edge is the assumed sensor employed for feedback. The wing is mounted to a device which is designed to permit motion in principally two modes - pitching and vertical translation (plunge).

The vibration frequencies, computed from a NASTRAN model of the BACT, are 3.4 Hz (plunge) and 5.2 Hz (pitch). Structural damping is assumed to be zero.

There are two CAP-TSD[21] aerodynamic representations of the BACT wind-tunnel model used in this paper: the first one is a 3-D model and the second one is an equivalent 2-D model. Most of the control law design and evaluation process results will be demonstrated with the 2-D model. However, the exact same process would be applied to the 3-D case. The results will begin with some basic steady aerodynamic data and then proceed to uncontrolled flutter calculations, and finally to controlled flutter calculations.

3.1 Basic aerodynamic and uncontrolled flutter results

In order to calculate rigid aerodynamic pressures, the fluid flow equation (TSD potential equation) is integrated in time by CAP-TSD without coupling structural dynamics equations. Compared with experimental results, the CAP-TSD results capture the location and strength reasonably well.

Overprediction of control surface forces is typical for inviscid codes. Therefore, care should be used in real applications to account for this overprediction when using an inviscid code.

For dynamic aeroelastic analyses in a computational aeroelasticty environment, two steps are employed in performing the calculations. In the first step, a static aeroelastic deformation is calculated to provide the initial flowfield for the dynamic aeroelastic solution. The dynamic solution is a perturbation about a converged static aeroelastic solution for each Mach number and dynamic pressure of interest. This method results in convergence of the steady-state plunge and pitch displacements, velocities, and accelerations. Once a static aeroelastic solution is computed, the next step is to prescribe either an initial condition on the displacements or velocities or an external input. For flutter calculations, initial conditions on the velocities are used to begin the dynamic structural integration. The ERA system ID method of (Juang, Papa)[19] was employed to estimate modal dampings and frequencies at various values of dynamic pressure. The CAP-TSD 3-d linear model shows approximately a 9% greater flutter dynamic pressure than the Doublet Lattice linear model. The Doublet Lattice linear model shows approximately a 3% lower flutter dynamic pressure than the experimental value. The CAP-TSD linear model shows approximately a 6% greater flutter dynamic pressure than the experimental value.

The CAP-TSD model shows approximately a 2% greater flutter dynamic pressure than the experimental value. In order to perform a system identification (ID), an exponential pulse provided a good input signal for identifying an equivalent linear model of the CAP-TSD outputs. The 8-state system ID model provides a very good representation of the CAP-TSD results for both plunge and pitch responses over the entire time history, for the 5 Hz case.

Figure 1 shows the chordwise pressure at the 60% span location at $M = 0.77$ for 2 degree angles of attack. CAP-TSD results with experimental results[20] and linear CAP-TSD results are also shown. At 2 degrees angle of attack, a weak shock near 20% chord has developed indicative of transonic flow. At higher degrees angles of attack, the shock moves aft and becomes moderately strong. Both the CAP-TSD and linear CAP-TSD results compare very well with experimental data aft of the 40% chord on the upper surface and compare well along the chord for the lower surface.

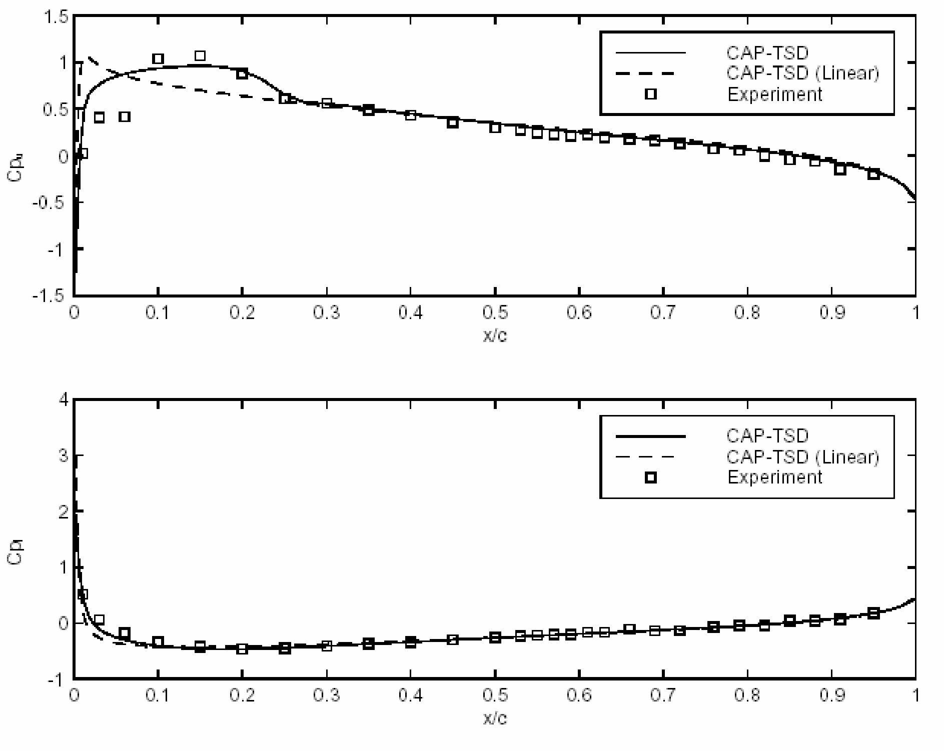

Figure 1: Aerodynamic results at $\alpha = 2$ degrees, M = 0.77, 60% semi-span

3.2 Control law design and evaluation

Five different cases have been investigated to illustrate the design methodology for a Mach number of 0.77 and a dynamic pressure of 5.75 kPa. The cases begin with the linear case at 0 degrees angle of attack and then proceed with nonlinear cases at 0 degrees angle of attack, 0.3 degrees angle of attack, and 0.6 degrees angle of attack. For each angle of attack, a system-ID model is developed and the control law designed with the system ID model for that angle of attack is compared with the control law for the linear case. The comparison is in terms of gain and phase margins, acceleration time history (using the exponential control input), and control surface deflection. In addition, a comparison of the eigenvalues for both the uncontrolled and controlled results for each case is presented. The control law designed with the linear model is intended to represent a control law using the state-of-the-art methodology. The primary design goal for all of the control law designs is to increase the damping of the system while exhibiting at least 6 dB gain margins and 60 degrees phase margins. The gain and phase margins are to account for uncertainty in the model. In addition, the control surface displacements, due to the feedback command, should be less than 1 degree in order to stay within a somewhat linear range of the control surface displacement. In each of the LQG designs, the weighting on the output and input during the regulator design and the intensities of the noise matrices during the Kalman filter design were varied by trial and error until a design that met the goal was determined.

Only last case will be described here. Case 5 employs the system ID model derived from the nonlinear CAP-TSD outputs at 0.6 degrees angle of attack to design a control law. Figure 2 shows a comparison of CAP-TSD outputs for the uncontrolled and controlled case. There is a significant reduction in the acceleration response with the controlled case. A gain margin of -9.66 dB and a phase margin of 64.17 degrees were determined using a Bode plot (not shown) of the open-loop system.

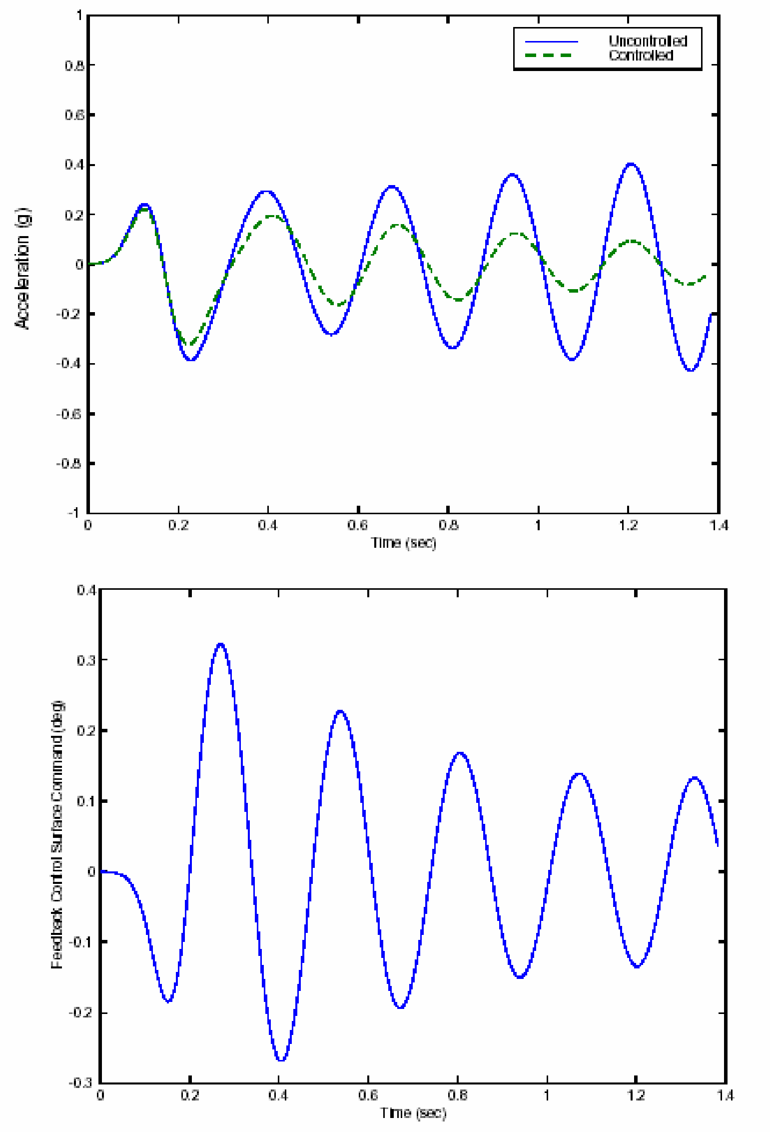

Figure 2: Case 5 controlled results

Figure 2 also shows the feedback control surface command for Case 5. Similar to the previous cases, the maximum control surface displacement is approximately 0.3 degrees and occurs during the exponential pulse excitation. When using this control law clearly show much better results, in particular stability margins and damping, than using the control law designed using the system ID model of the linear CAP-TSD outputs.

4. Conclusions

Equivalent linear models developed by employing a system identification technique can represent the input-output relationship of a computational aeroelasticity simulation very well.
For the BACT model used in this study, the system ID model represents the input-output relationship very well until the transonic flow conditions cause the shock on the upper surface to move aft of the 40% chord. At this point, extreme care must be used to obtain a good system ID model. A control law designed using a system ID model developed from a nonlinear simulation can control the nonlinear model better than a control designed using a system ID model developed from a linear computational aeroelasticity simulation.

Aknowledgements

This research was partially supported by the National research contract CEEX 05-D11-44/2005 and for attending the EUCASS Conference it was received partially support from ISTC. Sincerely thanks Dr. J.R. Newsom for some computational and experimental results

References

[1] Batina, J. T., Yang, T. Y. Transonic Calculation of Airfoil Stability and Response With Active Controls, AIAA 84-0873, April 1984.
[2] Friedman, P. P. The Renaissance of Aeroelasticity and Its Future, SDM *Lecture*, 38th AIAA/ASME/ASCE/AHS/ASC *Structures, Structural Dynamics and Materials Conference*, April 1997.
[3] Noll, T. E. Flight-Vehicle Materials, Structures, and Dynamics - Assessment and Future Directions, 5:179-212, 1993.
[4] Dowell, E. H. Nonlinear Aeroelasticity, AIAA-90-1031-CP, April 1990.
[5] Bennett, R. M. and Edwards, J. W. An Overview of Recent Developments in Computational Aeroelasticity, AIAA 98-2421, June 1998.
[6] Guruswamy, G. P., Tu, E. L., Goorjian, P. M. Transonic Aeroelasticity of Wings with Active Control Surfaces, AIAA 87-0709-CP, April 1987.
[7] Ominsky, D., Ide, H. An Effective Flutter Control Method Using Fast, Time-Accurate CFD Codes, AIAA 89-3468-CP, April 1989.
[8] Bendiksen, O., Hwang, G., Piersol, J. Nonlinear Aeroelastic and Aeroservoelastic Calculations for Transonic Wings, AIAA 98-1898, April 1998.
[9] Stephens, C, Arena, A., Gupta, K. CFD Based Aeroservoelastic Predictions with Comparisons to Benchmark Experimental Data, AIAA 99-0766, January 1999.
[10] Djayapertapa, L., Allen, C. B. Aeroservoelastic Computations in Unsteady Transonic Flow, AIAA 2000-4226, August 2000.
[11] Guillot, D. M., Friedmann, P. P., A Fundamental Aeroservoelastic Study Combining Unsteady CFD With Adaptive Control, AIAA 94- 1721-CP, April 1994.
[12] Friedmann, P. P., Guillot, D. M., Presente, E. Adaptive Control of Aeroelastic Instabilities in Transonic Flow and Its Scaling, AIAA 97-0581-CP, January 1997.
[13] Batina, J. T., Seidel, D. A., Bland, S. R., Bennett, R. M. Unsteady Transonic Flow Calculations for Realistic Aircraft Configurations, *Journal of Aircraft*, 26:21-22, 1989.
[14] Batina, J. T. Efficient Algorithm for Solution of the Unsteady Transonic Small-Disturbance Equation, *Journal of Aircraft*, 25:598-605, 1988.
[15] Juang, J. An Overview of Recent Advances in System Identification, AIAA-93-1664, April 1993.
[16] Phan, M. Q., Horta, L. G., Juang, J., Longman, R. W. Identification of Linear Systems by an Asymptotically Stable Observer, NASA TP 3164, June 1992.
[17] Newsom, J.R. Designing Active Control Laws in a Computational Aeroelasticity Environment, PhD Dissertation, Virginia Tech, 2002.
[18] Chen, W., Valasek, J. Observer/Kalman Filter Identification for On-line System Identification of Aircraft, AIAA-99-4173, August 1999.
[19] Juang, J.N., Pappa, R.S. An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction, *Journal of Guidance, Control, and Dynamics*, 8:620-627, 1985.
[20] Scott, R. C., Hoadley, S. T., Wieseman, C. D., Durham, M. H. The Benchmark Active Controls Technology Model Aerodynamic Data, AIAA 97-0829-CP, January 1997.
[21] Silva, W. A., Bennett, R. M. Investigation of the Aeroelastic Stability of the AFW Wind-Tunnel Model Using CAP-TSD, NASA TM 104142, September 1991.

This page has been purposedly left blank