JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 24, No. 1, January–February 2001

Generalized Predictive Control for Active Flutter Suppression

Pam Haley*
NASA Langley Research Center, Hampton, Virginia 23681
and
Don Soloway†
NASA Ames Research Center, Moffett Field, California 94035

Experimental results of a transonic wind-tunnel test that demonstrate the use of generalized predictive control for flutter suppression in a subsonic wind-tunnel wing model are presented. The generalized predictive control algorithm is based on the minimization of a suitable cost function over finite costing and control horizons. The cost function minimizes not only the sum of the mean square output of the plant predictions, but also the weighted square rate of change of the control input with its input constraints. An additional term was added to the cost function to compensate for dynamics of the wing model that cause it to be invariant to low input frequencies. This characteristic results in a control surface that drifts within the specified input constraints. The augmentation to the cost function that penalizes this low-frequency drift is derived and demonstrated. The initial validation of the controller uses a linear plant predictor model for the computation of the control inputs. Simulation results of the closed-loop system that were used to determine nominal ranges for the tuning parameters are presented. The generalized predictive controller based on the linear predictor model successfully suppressed the flutter for all testable Mach numbers and dynamic pressures in the transonic region in both simulation and wind-tunnel testing. The results confirm that the generalized predictive controller is robust to modeling errors.

Introduction

TYPICALLY, an aircraft's speed and altitude are limited by an envelope that is defined conservatively to be below the flutter boundary for that wing. This envelope is designed to keep the aircraft in flight conditions in which flutter does not occur. Future aircraft may incorporate flexibility in the structural design to enhance an aircraft's performance, increase its efficiency, or to reduce its weight and cost. An aircraft that incorporates flexibility in its design is likely to require an active flutter suppression (AFS) system to remove aeroelastic instabilities.

The benchmark active controls technology (BACT) subsonic wind-tunnel wing model is one of the models in the benchmark models program (BMP) at NASA Langley Research Center. The BMP includes a series of models, varying in complexity, that are used to study different aeroelastic phenomena and to validate different active controls techniques.[1] The dynamics of the BACT wind-tunnel model are such that the aeroelastic instabilities are relatively benign, thus making it simpler to build safety mechanisms into the test facility to take over when a controller fails without destruction of the model. This makes the BACT wind-tunnel model an excellent candidate for testing new control techniques. Some of the interesting aeroelastic challenges exhibited by the BACT model are classical transonic flutter, shock-induced instabilities, and separation-induced oscillatory instabilities. The results presented in this paper consider the classical transonic flutter problem only.

One type of controller tested during the wind-tunnel tests was a generalized predictive controller. Generalized predictive control (GPC) is a linear controller that the literature claims can control non-minimum phase plants, open-loop unstable plants, plants with variable or unknown dead time and that can systematically take into account real plant constraints in real time.[2] GPC is robust with respect to modeling errors, over- and underparameterization, and sensor noise.[2] An enhancement to GPC that takes into account plant nonlinearities is the neural generalized predictive controller (NGPC). NGPC bases its control laws on a nonlinear neural network model of the plant instead of a linear model. The validation of GPC for active flutter suppression presented in this paper is the beginning of a series of tests to verify the capabilities of NGPC. Another type of predictive control was also demonstrated during this set of wind-tunnel tests. The results of that work can be seen in Ref. 3.

In the next section the experimental setup will be described, followed by the following sections: GPC, BACT Modeling and Analysis, Augmentation to the Cost Function, Simulation Results, Wind-Tunnel Results, and Summary.

Experimental Setup

The wind-tunnel test was conducted in the transonic dynamics tunnel (TDT) at NASA Langley Research Center. The TDT is capable of controlling Mach and dynamic pressure independently over a range of values.[4] The BACT wing is a rigid rectangular wing with a NACA 0012 airfoil section. It is equipped with three control surfaces (trailing-edge flap, upper spoiler, and lower spoiler) that are positioned by hydraulic actuators. Linear accelerometers are located one at each corner of the wing, and they are used as the primary sensors for feedback control. The wing is mounted on a device called the pitch and plunge apparatus (PAPA), which is designed for rotation (pitching) and vertical translation (plunging) degrees of freedom.[5,6] The characteristics of the wing and its aeroelastic properties can be set by adjustments to the PAPA mount. The BACT wing and the PAPA mount together will be referred to as the BACT plant. A wiring diagram of the BACT control system is shown in Fig. 1.

The accelerometer signals are passed through a bank of 30-Hz antialiasing filters. These signals are sent to the digital controller after being sampled at 200 Hz and digitized by the 12-bit analog-to-digital converter. The digital controller produces command signals that are sent to the digital-to-analog converter. These signals command the positions of the control surfaces such that minimal accelerations are produced, thus suppressing flutter. The digital controller was implemented on a Pentium Pro 150 MHz personal computer and the data acquisition system was the Data Translation 2839 DAQ board. Even though a Pentium Pro processor was used during the wind-tunnel tests, the control algorithm only required 4% of the CPU. This number does not include data collection and data recording. For more information on the software implementation and timing specifications, refer to Ref. 7.

Received 7 October 1999; revision received 1 September 2000; accepted for publication 5 September 2000.

*Aerospace Technologist, Dynamics and Control Branch/Airborne Systems Competency, MS 132; p.j.haley@larc.nasa.gov.

†Computer Engineer, Nero Engineering and Smart Systems/Computational Sciences Division, MS 269-1; don@infi.net.

During the wind-tunnel tests a single-input/single-output (SISO) GPC system was used because the multi-input/multi-output (MIMO) implementation had not been completely developed. The results reported in this paper use the inboard trailing-edge accelerometer measurements as the feedback control signal and the position of the trailing-edge flap as the control input. The BACT plant displays two aeroelastic properties, pitch and plunge. The plunge mode contributes more to the amplitude at the flutter frequency than does the pitch mode. Therefore, the inboard trailing-edge accelerometer paired with the trailing-edge flap is more capable of suppressing plunge.

GPC

GPC belongs to the class of model-based predictive control (MBPC) strategies and was introduced by Clarke et al. in 1987.[2,8,9] MBPC techniques have been analyzed and implemented successfully in process control industries since the end of the 1970s and have continued to gain popularity with the increasing computational capability of computers.

The block diagram of the GPC system used in these experiments is shown in Fig. 2. It consists of four components: the BACT plant, a model for prediction, a reference signal that specifies the desired performance objective, and the cost function minimization (CFM) algorithm that calculates the control surface position command needed to produce the desired performance objective.

The GPC algorithm operates in two modes, prediction and control. Prediction occurs between samples by setting a double-pole/double-throw switch S to the plant model. The GPC algorithm utilizes the model to predict, over some finite horizon, the response of the plant to the inputs calculated by the CFM algorithm. The CFM algorithm minimizes a user-specified cost function to calculate the next control input. The GPC system is set back to a mode of control before the next sample time when the switch is set back to the BACT. At this time the control input that minimizes the cost function over the entire horizon is passed to the BACT as the position command $u(n)$. The algorithm used to accomplish this is outlined hereafter.

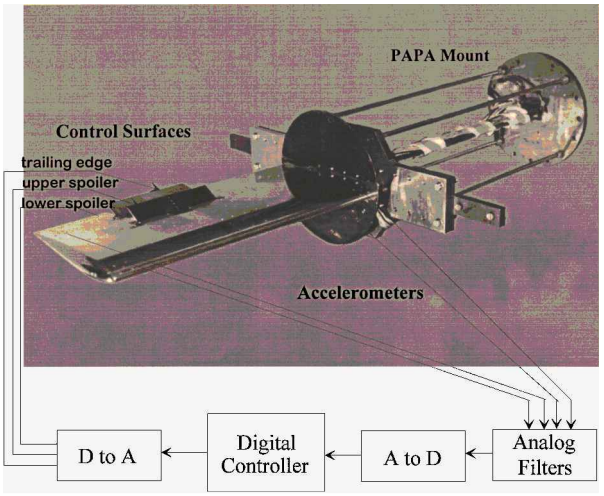

Fig. 1 Wiring of BACT control system.

The main steps of the GPC algorithm are as follows:

1) Starting with the previously calculated control input $u(n)$, predict the performance of the plant for the specified horizon using the model. The value of the horizon is determined through a priori tuning.

2) Calculate a new control input that minimizes the cost function.

3) Repeat steps 1 and 2 until desired minimization is achieved.

4) Send the best control input to the BACT as the new $u(n)$.

5) Repeat for each time step.

The cost function used for acceleration control of the BACT system, Eqs. (1) and (2), has three terms. The first term represents the sum of the mean square output of the BACT model. The model serves to predict the plant outputs from N_1 to N_2 future time steps. The second term is the weighted square of the control increments. The weighting factor λ_u acts to smooth the control inputs. The calculated inputs for the plant predictions form the control increments. They are calculated for N_u future time steps. The only constraint on the values of the horizons is that N_u and N_1 be less than or equal to N_2.

$$J = \sum_{j=N_1}^{N_2} [yn(n+j)]^2 + \sum_{j=1}^{N_u} \lambda_u(j)[\Delta u(n+j)]^2 + g(u) \qquad (1)$$

where N_1 is the minimum-costing horizon, N_2 is the maximum-costing horizon, N_u is the control horizon, yn is the predicted output of the model, λ_u is the control input weighting factor, and $\Delta u(n+j) = u(n+j) - u(n+j-1)$ and where

$$g(u) = \sum_{j=1}^{N_u} \left(\frac{s}{u(n+j) - \text{lower}} + \frac{s}{\text{upper} - u(n+j)} - \frac{4s}{\text{upper} - \text{lower}} \right) \qquad (2)$$

The third term of the summation of J, Eq. (2), defines constraints placed on the control input over a horizon of N_u. The constraint function $g(u)$ is plotted in Fig. 3. The first two terms of $g(u)$ form the two sides of the function, and the third part ensures that the minimum of the function is zero.

The sharpness s controls the shape of the constraint function, and the sides are bounded by upper and lower, the input constraint. The smaller the value of s, the sharper the corners get. In practice, s is set to a very small number, such as 10^{-20}. Note from both the equation and Fig. 3 that as the control input u approaches either the upper or lower bound, the value of the input constraint function approaches infinity, which places a high cost on the minimization of this term. If the minimization produces either $u \leq$ lower or $u \geq$ upper, then u is set to $u + \varepsilon$ or $u - \varepsilon$, respectively. The value of ε is set to 10^{-6} (Ref. 10).

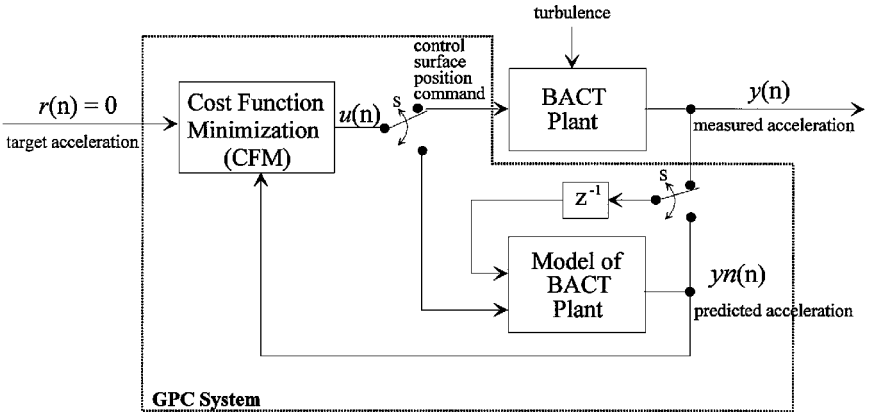

Fig. 2 Block diagram of the GPC system.

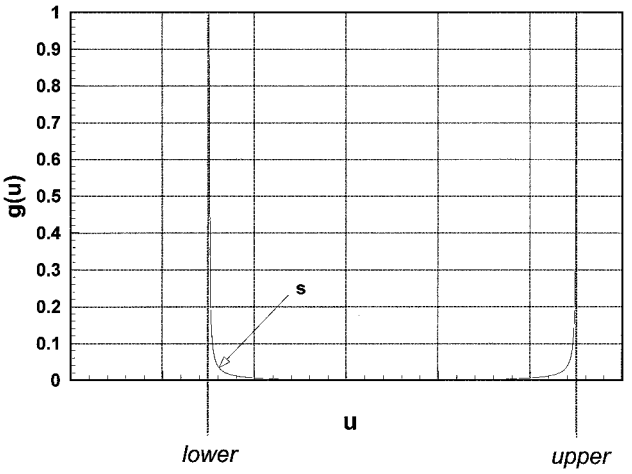

Fig. 3 Plot of input constraint function.

Currently, there is no systematic way to determine the values for the four tuning parameters N_1, N_2, N_u, and λ_u, for a nonlinear system. For a linear system the necessary condition to achieve stability is N_u equal to the order of the system whereas N_2 is used to tune for performance. The value for λ_u should approach zero. Dead time in the system is accounted for with the value of N_1.

The CFM algorithm used to minimize the cost function is the Newton–Raphson iterative algorithm. Newton–Raphson is a quadratically converging algorithm that requires the calculation of the Jacobian and the Hessian. Although the Newton–Raphson algorithm can be computationally expensive, the low number of iterations needed for convergence makes it a feasible algorithm. A complete derivation of the GPC algorithm for a general SISO system is developed in Ref. 7. (The original derivation of the cost function was developed for tracking control of nonlinear plants using a neural network for the plant's model. The equations for the neural model can still be used for a linear model if the neural network's activation function is made linear, thus making the plant's model a linear ARMA model.) The computational issues of Newton–Raphson are also addressed in Ref. 7.

BACT Modeling and Analysis

The GPC algorithm uses the output of the model to predict the BACT plant dynamics to an arbitrary input. With an adequate model and the correct tuning of the control parameters (N_1, N_2, N_u, and λ_u) the inboard trailing-edge accelerometer may be regulated to zero g. Because no systematic procedure exists to determine the values of these tuning parameters, the tuning of the controller can be quite cumbersome. This process could be especially difficult if tuning were to occur during real-time control because each wrong choice could result in instability of the system. For this reason, a GPC simulation was performed to determine the nominal ranges for the control parameters for the BACT plant. The BACT plant block, in Fig. 2, was simulated using a linear model that was developed previously from the knowledge of the plant and system identification techniques using preexisting wind-tunnel data.[11] The model of the BACT plant was a reduced-order discrete model based on the model obtained in Ref. 11. (There were two reasons for reducing the order of the plant's model. First, to develop a realistic simulation, the model needs to contain unmodeled dynamics. Second, in receding horizon control (which GPC is a type) typically the computational complexity increases with the order of the plant; therefore, the smallest acceptable model is desirable). The sampling frequency for the discretization was 200 Hz. Both models were developed for flight conditions below the flutter boundary, making them open-loop stable. The magnitude and phase plots of these models are shown in Figs. 4 and 5, respectively.

The reduced-order model captures the dominant modes and the dynamics near the flutter frequency at 4 Hz, but (as seen in Figs. 4 and 5) the higher-order terms have been left unmodeled. The flutter instability occurs at a dynamic pressure $\bar{q}$ above 150 psf and at a frequency of around 4 Hz. Figures 4 and 5 also indicate that both

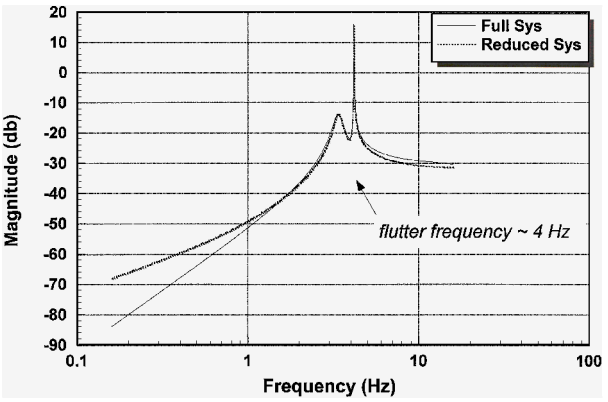

Fig. 4 Frequency response of the BACT models.

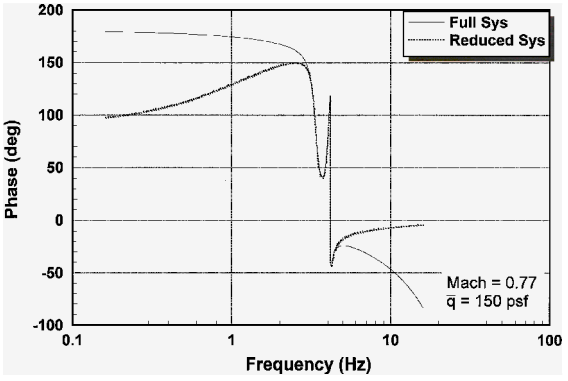

Fig. 5 Phase plot of BACT models.

models have at least one zero at the origin. This implies that the dc component and very low frequencies effectively do not pass through the plant, making the acceleration output invariant to a slow drift in the position of the control surface. This characteristic of the plant affects the performance of the GPC algorithm and is not accounted for in the original cost function in Eqs. (1) and (2). The result is a drift in the control surface position within the specified input constraints. This problem is also confirmed in the wind-tunnel test. The solution was to augment the cost function with a term that penalizes a drift in the control surface position. This enhancement is developed in the next section, along with the derivation of the Jacobian and Hessian needed for the cost function minimization. This derivation augments the derivation found in Ref. 7.

Augmentation to the Cost Function

As mentioned in the preceding section, a low-frequency drift in the control input was experienced during wind-tunnel testing. To correct this outcome, the cost function of Eqs. (1) and (2) was augmented with a frequency weighted cost on the control input. The new cost function is given by

$$J = \sum_{j=N_1}^{N_2} [yn(n+j)]^2 + \sum_{j=1}^{N_u} \lambda_u(j)[\Delta u(n+j)]^2 + g(u) + \sum_{j=1}^{N_u} \lambda_f(j) uf(n+j)^2 \tag{3}$$

where the weighted control input is the output of a discrete-time filter of the form

$$uf(n) = \frac{b_0}{a_0}u(n) + \sum_{k=1}^{d}\left(\frac{a_k}{a_0}uf(n-k) + \frac{b_k}{a_0}u(n-k)\right)$$

and $\lambda_f(j)$ is a scalar to balance the contribution of the new term. The discrete filter $uf(n)$ is designed to amplify the frequencies that are to be penalized when minimizing the cost function. The design approach was to design a continuous-time high-pass filter, discretize it,

and then invert the zero/pole dynamics. The resulting filter amplifies very low frequencies, and the cost function minimizes them.

To include this filter in the derivation of the CFM iterative solution found in Ref. 7, the Jacobian and the Hessian of the filter are needed. Looking just at the filter part of the cost function, let

$$Jf = \sum_{j=1}^{N_u} \lambda_f(j) uf(n+j)^2 \tag{4}$$

The calculation of the elements of the Jacobian are found by evaluating

$$\frac{\partial Jf}{\partial u(n+h)} = 2\sum_{j=1}^{N_u} \lambda_f(j) uf(n+j) \frac{\partial uf(n+j)}{\partial u(n+h)} \tag{5}$$

and the elements of the Hessian are found by evaluating

$$\frac{\partial^2 Jf}{\partial u(n+h)\partial u(n+m)} = 2\sum_{j=1}^{N_u} \lambda_f(j) \left(\frac{\partial uf(n+j)}{\partial u(n+h)} \frac{\partial uf(n+j)}{\partial u(n+m)} + uf(n+j) \frac{\partial^2 uf(n+j)}{\partial u(n+h)\partial u(n+m)} \right)$$

where h and m equal from 1 to N_u.

Because the filter $uf(n)$ is linear, its second derivative is equal to zero. This reduces the Hessian to

$$\frac{\partial^2 Jf}{\partial u(n+h)\partial u(n+m)} = 2\sum_{j=1}^{N_u} \lambda_f(j) \frac{\partial uf(n+j)}{\partial u(n+h)} \frac{\partial uf(n+j)}{\partial u(n+m)} \tag{6}$$

To solve Eqs. (5) and (6), the first derivative of the filter with respect to the input is derived and results in

$$\frac{\partial uf(n+j)}{\partial u(n+h)} = \frac{b_0}{a_0} \frac{\partial u(n+j)}{\partial u(n+h)} + \sum_{k=1}^{d} \left(\frac{a_k}{a_0} \frac{\partial uf(n-k+j)}{\partial u(n+h)} + \frac{b_k}{a_0} \frac{\partial u(n-k+j)}{\partial u(n+h)} \right) \tag{7}$$

The first term of Eq. (7) and the second term in the summation are expanded and reduced by eliminating all derivatives equal to zero. The resulting simplifications are combined to form the conditional equation

$$\frac{\partial uf(n+j)}{\partial u(n+h)} = \sum_{k=1}^{d} \frac{a_k}{a_0} \frac{\partial uf(n-k+j)}{\partial u(n+h)} + \begin{cases} b_{j-h}/a_0 & 0 \le j-h \le d \\ 0 & \text{else} \end{cases} \tag{8}$$

Equations (5) and (6) should be added to the Jacobian and Hessian equations of Ref. 7 for a complete solution to the GPC control input.

Simulation Results

The GPC simulation described in the BACT plant analysis section was used to determine the nominal ranges for the control parameters before the wind-tunnel testing. The numerical model and the reduced-order model were developed for a Mach number of 0.77 and dynamic pressure of 150 psf, a flight condition that is below the flutter boundary. The reduced-order model, which is in the form of an autoregressive moving-average (ARMA) model, is represented by

$$\frac{Z\{yn(n)\}}{Z\{u(n)\}} \equiv \frac{b_0 - b_1 z^{-1} + b_2 z^{-2} - b_3 z^{-3} + b_4 z^{-4}}{1 - a_1 z^{-1} + a_2 z^{-2} - a_3 z^{-3} + a_4 z^{-4}}$$

where $b_0 = 0.025256, b_1 = 0.099239, b_2 = 0.14661, b_3 = 0.096511$, $b_4 = 0.023884$, $a_1 = 3.9603$, $a_2 = 5.9097$, $a_3 = 3.9379$, and $a_4 = 0.98868$.

Because there is no systematic procedure for selecting the values of the control parameters, several experiments were conducted to find a set of control parameters that produce the smallest rms acceleration around the flutter frequency. The four parameters N_1, N_2, N_u, and λ_u took on the combinations of the values as follows: $N_1 = 1$, N_2, $N_u \in \{1, 2, 3, 4\}$ (such that $N_2 \ge N_u$), and $\lambda_u \in \{0.0001, 0.001, 0.01, 0.1\}$. The smallest rms value occurred when $N_1 = 1$, $N_2 = 2$, $N_u = 1$, and $\lambda_u = 0.01$.

For these simulations the relative magnitude of the control signal was constrained to ± 3 deg to ensure that the controller did not produce large deflections in the trailing-edge flap. Large deflections for flutter suppression should be avoided because the control surfaces may have physical constraints, and larger deflections are typically reserved for flight control. The physical limitations in the deflection of the control surface can be incorporated in the cost function by setting the input constraint parameters accordingly. To handle this constraint, the parameters s, upper, and lower are set to 10^{-20}, 3, and -3, respectively, in the third summation of Eq. (2).

Figure 6 shows the commanded deflection of the trailing-edge flap, and Fig. 7 shows a comparison of the open- and closed-loop response. From Fig. 7, the closed-loop frequency response shows that this set of control parameters has attenuated the flutter by approximately 17 dB. This reduction is acceptable for flutter suppression of the BACT plant and is similar to other SISO controllers tested. For this simulation, the filter portion of the augmented cost function was not used so that the low-frequency drift in the control signal (Fig. 6) can be seen. The effectiveness of the filter for removing the low-frequency drift can be seen in Fig. 8.

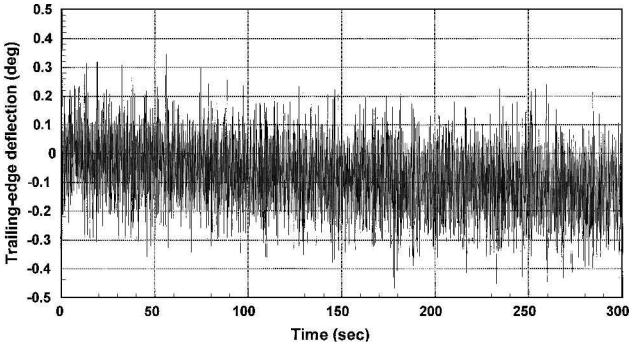

Fig. 6 Commanded control surface deflection without filter.

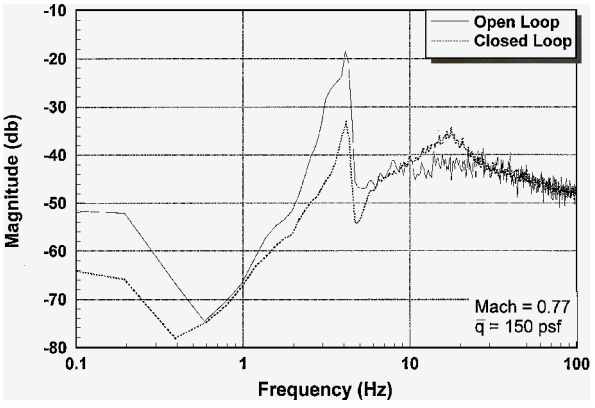

Fig. 7 Frequency response of trailing-edge accelerometer.

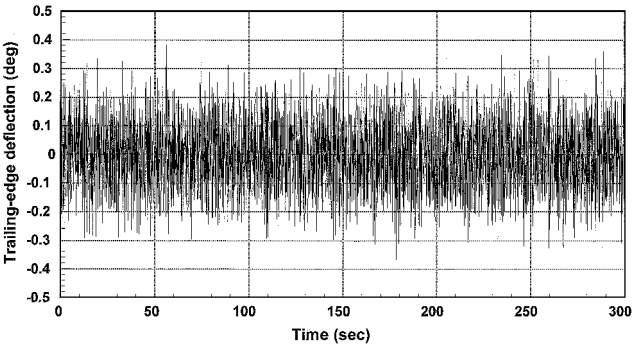

Fig. 8 Commanded control surface deflection using a filter.

The filter design started with a washout filter with the transfer function

$$f(s) = [s/(s+1)]u(s)$$

The washout filter was discretized with a sampling time of 0.005 s using a step-invariant transform resulting in

$$f(z) = \frac{1 - z^{-1}}{1 - 0.995012z^{-1}}u(z)$$

Inverting the filter we have

$$uf(z) = \frac{1 - 0.995012z^{-1}}{1 - z^{-1}}u(z)$$

To add this filter to the cost function, set $d = 1$, and then set the coefficient parameters to $a_0 = 1$, $a_1 = -1$, $b_0 = 1$, and $b_1 = -0.995012$. The filter's weighting factor λ_f was set to 0.0001. Using this filter in the simulation yields the drift free control signal in Fig. 8. The relative magnitude of the control signal has been left unchanged. The frequency response shown in Fig. 9 shows that the filter has little effect on the closed-loop response.

This simulation was also tested for flight conditions above flutter. The dynamic pressure was varied from 150 psf up to 250 psf with Mach remaining the same at 0.77. All simulations showed similar flutter suppression capability using the same GPC system, even though the GPC system was developed for flight conditions below flutter. Therefore, the simulation results demonstrated that a fixed GPC algorithm with input constraints can provide flutter suppression for a relatively wide range of flight conditions. The robustness properties of the GPC algorithm are also confirmed with the wind-tunnel test.

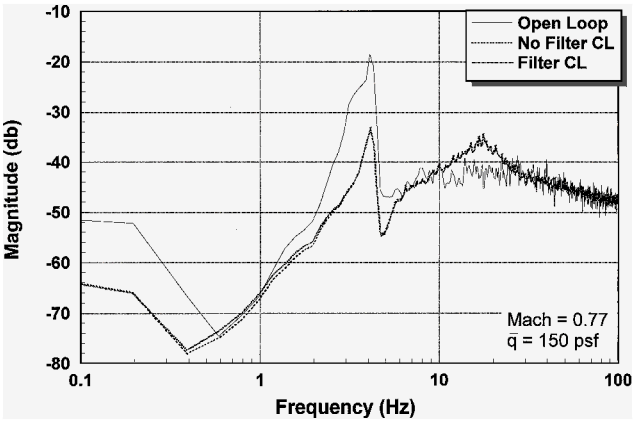

Fig. 9 Frequency response of trailing-edge accelerometer.

Wind-Tunnel Results

During the wind-tunnel testing, the same fourth-order linear ARMA model that was used during simulations was used as the model for GPC predictions. It was found that the same values for the control parameters in simulation also were the best values for control of the actual BACT plant.

With the fixed GPC, the closed-loop system had desirable performance characteristics. Test results without frequency weighting and control inputs constrained to ± 3 deg are given in Figs. 10 and 11. The curve in Fig. 10 indicates that the input constraints are satisfied and that the command signal has the low-frequency drift problem as expected. This data set represents conditions where Mach was varied from 0.75 to 0.79 and dynamic pressure was varied from 184 to 200 psf. The entire set of flight conditions were above the flutter boundary, and as seen in Fig. 11, the trailing-edge acceleration was maintained near zero throughout the test.

Test results with frequency weighting and input constraints set to ± 10 deg are given in Figs. 12 and 13. In this plot, Mach and dynamic pressure were maintained at 0.77 and 178 psf, respectively.

The time intervals of Fig. 12 with no commanded input are times when the control was allowed to go open loop. In the corresponding time intervals in Fig. 13, the accelerometer measurements start to grow. The longer the time period that the BACT plant remained uncontrolled, the larger the acceleration became and the larger the commanded deflection that was needed to regain control. Notice that the control input did not contain a low-frequency drift. The filter portion of the cost function was activated with λ_f equal to 0.0001. Because allowing a controller to go open loop during flight is not realistic, the commanded control input was set to plus and minus 10 deg to give the controller full dynamic range of the actuator.

There are several desirable control characteristics that need to be incorporated in the design of an AFS system. First, the closed-loop system must be robust to modeling errors. Next, the controller must also be able to dampen the flutter to some acceptable magnitude, within an allowable time period, and with minimal control surface deflections. In the case of the BACT plant, the damping time is not needed to be as short as with a high-performance wing. Also, here the commanded inputs were only used for flutter suppression. It would be desirable for the control surface deflections to be smaller if they were also used for flight control.

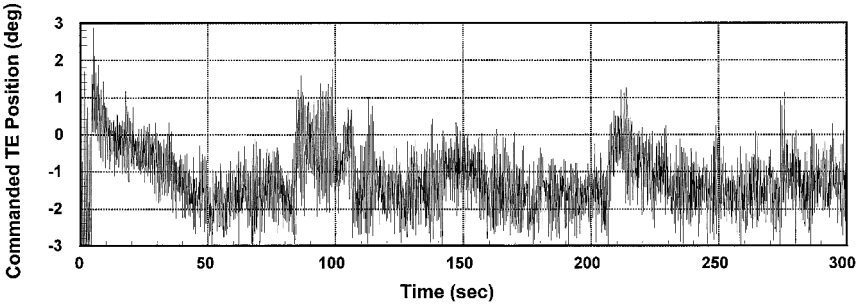

Fig. 10 Commanded trailing-edge position for varying flight conditions.

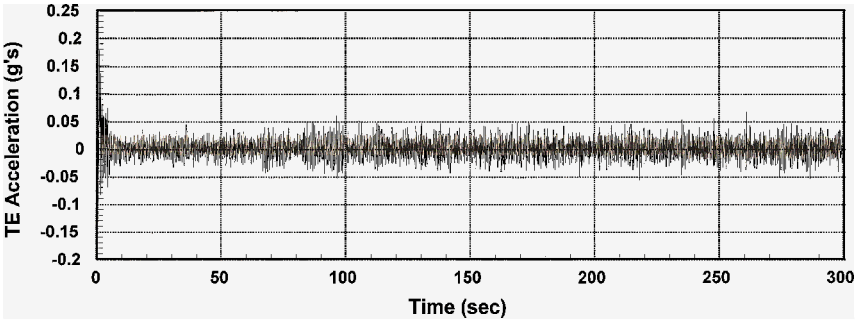

Fig. 11 Accelerometer measurements for varying flight conditions.

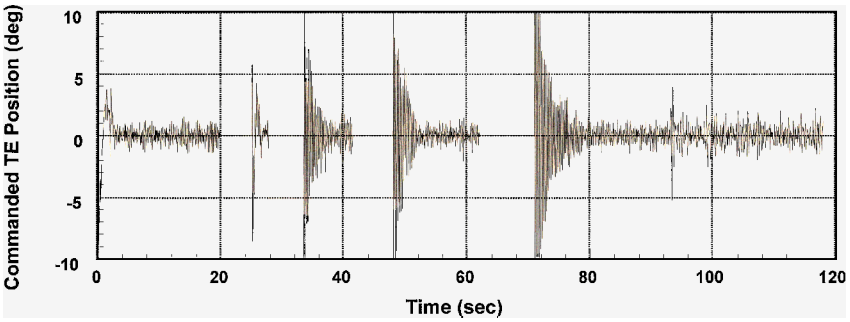

Fig. 12 Commanded trailing-edge position during open- and closed-loop control.

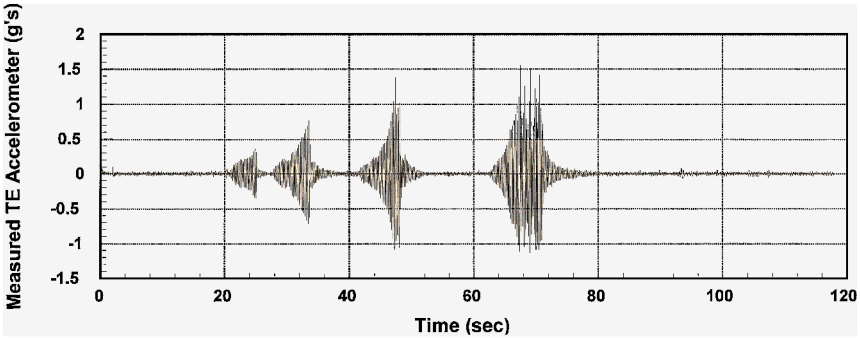

Fig. 13 Accelerometer measurements during open- and closed-loop control.

Summary

The results from the wind-tunnel test showed that for the tested flight conditions, the GPC was able to suppress flutter using a nominal linear model of the BACT plant. The wind-tunnel tests also verified that augmenting the cost function with frequency weighting on the control input is a feasible way of solving the controller's drift problem.

To increase the flutter suppression capability of the GPC two improvements are being considered. First, a nonlinear neural network model of the BACT plant is being developed. This would allow the GPC to make better predictions, thus improving performance: increased flutter damping in less time and with smaller control deflections. A neural-network model could also adapt to time-varying plant dynamics. A second improvement will be to use a MIMO implementation of the GPC. Using the two inboard accelerometer measurements for feedback and using the upper spoiler with the trailing-edge flap as command inputs will allow the GPC to control both plunge and pitch modes, leading to enhanced flutter suppression. These improvements could lead to a shorter damping time with less control surface movement and increase the capability of the GPC to control more difficult flight conditions.

References

[1]Scott, R. C., Hoadley, S., Wieseman, C., and Durham, M., "Benchmark Active Controls Technology Model Aerodynamic Data," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, 2000, pp. 914–921.

[2]Clarke, D. W., Mohtadi, C., and Tuffs, P. C., "Generalized Predictive Control—Part 1: The Basic Algorithm," *Automatica*, Vol. 23, No. 2, 1987, pp. 137–148.

[3]Scott, R. C., and Pado, L., "Active Control of Wind-Tunnel Model Aeroelastic Response Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 6, 2000, pp. 1100–1108.

[4]Aeroelasticity Branch Staff, "Langley Working Paper–The Langley Transonic Dynamics Tunnel," NASA Langley White Paper, LWP-799, NASA Langley Research Center, Hampton, VA, Sept. 1969.

[5]Rivera, J. A., 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," AIAA Paper 92-2396, April 1992.

[6]Rivera, J. A., Jr., Dansberry, B. E., Durham, M. H., Bennett, R. M., and Silva, W. A., "Pressure Measurement on a Rectangular Wing with a NACA 0012 Airfoil During Conventional Flutter," NASA TM-104211, July 1992.

[7]Soloway, D., and Haley, P., "Neural Generalized Predictive Control: A Newton–Raphson Implementation," Inst. of Electrical and Electronics Engineers, Paper ISIAC-TA5.2, Sept. 1996.

[8]Clarke, D. W., Mohtadi, C., and Tuffs, P. C., "Generalized Predictive Control—Part 2: Extensions and Interpretations," *Automatica*, Vol. 23, No. 2, 1987, pp. 149–160.

[9]Clarke, D. W., "Advances in Model-Based Predictive Control," *Advances in Model-Based Predictive Control*, edited by D. W. Clarke, Oxford Univ. Press, 1994.

[10]Soloway, D., "Neural Generalized Predictive Control for Real-Time Control," M.S. Thesis, Electrical and Computer Engineering, Old Dominion Univ., Norfolk, VA, Aug. 1996.

[11]Waszak, M. R., "Modeling the Benchmark Active Control Technology Wind-Tunnel Model for Application to Flutter Suppression," AIAA Paper 96-3437, July 1996.