Experimental Validation of Generalized Predictive Control for Active Flutter Suppression

Pamela Haley
Don Soloway
NASA Langley Research Center
Hampton, VA 23681
pjh@larc.nasa.gov
don@ptollem.arc.nasa.gov

Abstract
This paper presents a status report on the experimental results of the transonic wind-tunnel test conducted to demonstrate the use of Generalized Predictive Control for flutter control of a subsonic airfoil. The Generalized Predictive Control algorithm is based on the minimization of a suitable cost function over a finite prediction horizon. The cost function minimizes the sum of the mean square output of the plant predictions using a suitable plant model, weighted square of control increments, and the term which incorporates the input constraints. The characteristics of the subsonic airfoil are such that its dynamics are invariant to low input frequencies. This results in a control surface that drifts within the specified input constraints. An 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. The Generalized Predictive Controller based on this model could successfully suppress the flutter for all testable mach numbers and dynamic pressures in the transonic region. The wind-tunnel test results confirmed that the Generalized Predictive Controller is robust to modeling errors. The simulation results that were used to determine the nominal ranges for control parameters before wind-tunnel testing are also included. The wind-tunnel test results were in good agreement with the results of the simulation.

Introduction
The Benchmark Active Controls Technology (BACT) subsonic wing is one of the modeled airfoils in the Benchmark Models Program (BMP) at NASA Langley Research Center. The BMP includes a series of models varying in complexity which are used to study different aerelastic phenomena and to validate different active controls techniques. The dynamics of the BACT wing are such that the aerelastic instabilities are slow, 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 model an excellent candidate for testing new control techniques. Some of the interesting aerelastic challenges exhibited by the BACT model are the classical transonic flutter, shock induced instabilities, and separation induced oscillatory instabilities. The research presented in this paper confronts the classical transonic flutter problem only.

Typically, an aircraft's speed and altitude are limited by an envelope that is defined to be conservatively below the flutter boundary for that airfoil. This envelope is designed to keep the aircraft in flight conditions in which flutter is unlikely to occur. An aircraft that pushes this envelope is likely to require an active flutter suppression (AFS) system to remove aerelastic instabilities. An AFS system would facilitate an increase in the aircraft's performance and also permit flexibility in the structural design of the airfoil which could lead to a reduction in weight and cost.

Since flutter is a highly nonlinear phenomena an AFS system that takes into account this nonlinearity is likely to out perform a controller based on linear control laws. A controller used for AFS must be robust and be able to dampen the flutter to some acceptable magnitude, within an allowable time period, with minimal control energy. Generalized Predictive Control (GPC) is a linear controller that is known to be robust with respect to modeling errors. 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 is the beginning of a series of tests to verify the capabilities of NGPC.

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 in which flutter occurs. The BACT wing is a rigid rectangular wing with a NACA 0012 airfoil section. It is equipped with three control surfaces (trailing-edge, 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 to allow rotation (pitching) and vertical translation (plunging) modes. The characteristics of the wing and its aerelastic 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 as shown in Figure 1.

![Figure 1. Wiring of BACT control system](image)

1 For more information on the software implementation and timing specifications see [7].
Predictive Controller was used in a single-input single-output (SISO) mode. The command input specifies the position of the trailing-edge control surface and the accelerometer sensor measurements consists of the inboard trailing-edge accelerometer signal.

**Generalized Predictive Control**

One of the controllers evaluated during the wind-tunnel testing was a Generalized Predictive Controller (GPC). GPC was introduced by Clarke and his co-workers in 1987 and it belongs to a class of Model-Based Predictive Control (MBPC) [4][5][6]. MBPC techniques have been analyzed and implemented successfully in process control industries since the end of the 1970’s and continue to be used because they can systematically take into account real plant constraints in real-time. GPC is known to control non-minimum phase plants, open-loop unstable plants and plants with variable or unknown dead time. This controller is also robust with respect to modeling errors, over and under parameterization, and sensor noise [4].

The GPC system for the BACT plant is shown in Figure 2. It consists of four components: the BACT plant, a reference signal that specifies the desired performance, a model of the plant, and the Cost Function Minimization (CFM) algorithm that determines the control surface position command needed to produce the desired performance. The principal components of the GPC algorithm are the CFM and model blocks.

![Diagram](image)

**Figure 2. Block diagram of the GPC system and algorithm**

For the BACT plant the GPC was used in a regulator mode where the reference signal, \( r(n) \), was set to zero. The output of the CFM algorithm is either used as an input to the BACT plant or the BACT model. The double pole double throw switch, \( S \), is set to the BACT plant when the CFM algorithm has solved for the best input, \( u(n) \), that will minimize a specified cost function. Between samples, the switch is set to the model where the CFM algorithm is used to predict the BACT plant dynamics to an arbitrary input. With an accurate model and the correct tuning of the control parameters \( N_1, N_2, N_p, \) and \( \lambda \), the inboard trailing-edge accelerometer may be regulated to zero g’s. Since 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 occurs during real-time control because each wrong choice could result in the 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 Figure 2, was implemented successfully in process control industries since the end of the 1970’s and continue to be used because they can systematically take into account real plant constraints in real-time. GPC is known to control non-minimum phase plants, open-loop unstable plants and plants with variable or unknown dead time. This controller is also robust with respect to modeling errors, over and under parameterization, and sensor noise [4].

The cost function used for the BACT plant is

\[
J = \sum_{j=0}^{N_u} [y(n+j)]^2 + \sum_{j=0}^{N_u} \lambda_j |\Delta u(n+j)|^2 + \sum_{j=1}^{N_u} \left[ \frac{u(n+j) - \frac{r}{2} - \frac{r}{2} - b + \Delta u(n+j)}{2} \right]^4
\]

where \( N_1 \) is the minimum costing horizon, \( N_2 \) is the maximum costing horizon, \( N_p \) is the control horizon, \( y(n) \) is the predicted output of the model, \( \lambda \) is the control input weighting factor, \( \Delta u(n+j) \) the change in \( u \) and is defined as \( u(n+j) - u(n+j-1) \), \( r \) is the range of the constraint function, \( s \) is the sharpness of the constraint function, \( b \) is an offset to the range.

This cost function minimizes the sum of the mean square output of the plant predictions using a suitable plant model, weighted square of control increments, and the term which incorporates the input constraints.

When this cost function is minimized, a control input that meets the constraints is generated that regulates the measured acceleration to the specified range. There are four tuning parameters in the cost function, \( N_1, N_2, N_p, \) and \( \lambda \). The plant’s outputs are predicted from \( N_1 \) to \( N_2 \) future time steps. The bound on the control horizon is \( N_p \). The only constraint on the values of \( N_p, \) and \( N_1 \) is that these bounds must be less than or equal to \( N_2 \). The second summation contains a weighting factor, \( \lambda \), that is introduced to control the balance between the first two summations. The weighting factor, \( s \) characterizes the sharpness, range, and offset of the constraint function respectively. The sharpness, \( s \), controls the shape of the constraint function. The constraint function plot looks like the letter U. The smaller the value of \( s \), the sharper the corners get. In practice, \( s \) is set to a very small number, for example \( 10^{-6} \).

A complete derivation of the GPC algorithm for a general system is developed in [7]. The algorithm used to minimize the cost function is the Newton-Raphson iterative algorithm. Newton-Raphson is a quadratically converging algorithm which requires the calculation of the Jacobian and the Hessian. Although the Newton-Raphson algorithm is computationally expensive it is justified by the low number of iterations needed for convergence. The computational issues of Newton-Raphson are also addressed in [7].

**BACT Plant 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 \( (N_1, N_2, N_p) \) the inboard trailing-edge accelerometer may be regulated to zero g’s. Since 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 occurs during real-time control because each wrong choice could result in the 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 Figure 2, was implemented successfully in process control industries since the end of the 1970’s and continue to be used because they can systematically take into account real plant constraints in real-time. GPC is known to control non-minimum phase plants, open-loop unstable plants and plants with variable or unknown dead time. This controller is also robust with respect to modeling errors, over and under parameterization, and sensor noise [4].

2 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.
simulated using a linear model that was previously developed from the knowledge of the plant and system identification techniques using preexisting wind-tunnel data [8]. The model of the BACT plant was a reduced order discrete model based on the model obtained in [8]. The sampling frequency for the discretization was 200 Hertz. Both models were developed for flight conditions below the flutter boundary. The magnitude and phase plots of these models are shown in Figures 3 and 4 respectively.

To correct this problem, the cost function of eq. (1) was augmented with a frequency weighted cost on the control input. The new cost function is given by

\[ J = \sum_{i=1}^{N} (y[n+j])^2 + \sum_{j=1}^{N} \lambda_j/j \Delta u[n+j]^2 \]  

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

\[ u_f(n) = \frac{b_0}{a_0} u(n) + \sum_{j=1}^{N} \frac{a_j}{a_0} u(n-j) \]

and \( \lambda_j(j) \) is a scalar to balance the contribution of the new term. The discrete filter, \( u_f(n) \), 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 [7], the Jacobian and the Hessian of the filter are needed. Looking just at the filter part of the cost function, let

\[ J_f = \sum_{j=1}^{N} \lambda_j(j) u_f(n+j)^2 \]  

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

\[ \frac{\partial J_f}{\partial u(n+h)} = 2 \sum_{j=1}^{N} \lambda_j(j) u_f(n+j) \frac{\partial u_f(n+j)}{\partial u(n+h)} \]  

and the elements of the Hessian are found by evaluating

\[ \frac{\partial^2 J_f}{\partial u(n+h) \partial u(n+m)} = 2 \sum_{j=1}^{N} \lambda_j(j) \left[ \frac{\partial u_f(n+j)}{\partial u(n+h)} \frac{\partial u_f(n+j)}{\partial u(n+m)} + u_f(n+j) \frac{\partial^2 u_f(n+j)}{\partial u(n+h) \partial u(n+m)} \right] \]

where \( h \) and \( m \) equal 1 to \( N \).

Since the filter, \( u_f(n) \), is linear, its second derivative is equal to zero. This reduces the Hessian to

\[ \frac{\partial^2 J_f}{\partial u(n+h) \partial u(n+m)} = 2 \sum_{j=1}^{N} \lambda_j(j) \frac{\partial u_f(n+j)}{\partial u(n+h)} \frac{\partial u_f(n+j)}{\partial u(n+m)} \]

To solve eqns. (4) and (5) the first derivative of the filter with respect to the input is derived and results in

\[ \frac{\partial u_f(n+j)}{\partial u(n+h)} = \frac{b_0}{a_0} \frac{\partial u_f(n+j)}{\partial u(n+h)} + \sum_{j=1}^{N} \frac{a_j}{a_0} \frac{\partial u_f(n-k+j)}{\partial u(n+h)} \]

The first term of eq. (6) and the second term in the summation is expanded and reduced by eliminating all derivatives equal to zero. The resulting simplifications are combined to form the conditional equation

\[ \frac{\partial u_f(n+j)}{\partial u(n+h)} = \frac{\lambda_j(j) \Delta u(n+j)}{\sum_{i=1}^{N} a_i \frac{\partial u_f(n+i)}{\partial u(n+h)} + \frac{b_0}{a_0} \Delta u(n+h)} \]

Equations (4) and (5) should be added to the Jacobian and Hessian equations of [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 well below the flutter boundary. The reduced order model which is in the form of an auto-regressive moving-average (ARMA) model is represented by

\[ Z(y(n)) = \frac{0.025256 - 0.099239 z^{-1} + 0.146612 z^{-2} - 0.096512 z^{-3} + 0.023884 z^{-4}}{1 - 39603 z^{-1} + 59097 z^{-2} - 3.9379 z^{-3} + 0.98868 z^{-4}} \]

Since 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 \( N_x \) took on the combinations of the values as follows: \( N_1 = 1, \), \( N_2, N_u \in \{1, 2, 3, 4\} \) such that \( N_2 < N_u \) and \( \lambda_a = 0.0001, 0.001, 0.01, 0.1 \). The smallest RMS value occurred when \( N_1 = 1, N_2 = 2, N_u = 1 \) and \( \lambda_a = 0.01 \).

For these simulations the relative magnitude of the control signal was set to \( \pm 3 \) degrees to insure that the controller did not produce large deflections in the trailing-edge control surface. 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 \( b, s, \) and \( r \) are set to 0, 10\(^{-5}\), and 6 respectively in the third summation of eq. (2).

Figure 5 shows the commanded deflection of the trailing-edge control surface and Figure 6 shows a comparison of the open and closed loop response. From Figure 6, the closed-loop frequency response shows that this set of control parameters has attenuated the flutter by approximately 17 decibels. This reduction is acceptable for flutter suppression of the BACT plant. The control signal seen in Figure 5 shows a drift. In this simulation, the filter portion of the cost function was not activated to demonstrate the effectiveness of the filter for removing the low frequency drift.

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

\[ f(z) = \frac{1}{z+1} u(z) \]

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

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

Inverting the filter we have

\[ u_f(z) = \frac{1-0.995012 z^{-1}}{1-z^{-1}} u(z) \]

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, \( \lambda_f \) was set to 0.0001. Using this filter in the simulation yields the drift free control signal in Figure 7. The relative magnitude of the control signal has been left unchanged. The frequency response shown in Figure 8 shows that the filter has little effect.

![Figure 5. Commanded control surface deflection without filter](image)

![Figure 6. Frequency Response of trailing-edge accelerometer](image)

![Figure 7. Commanded control surface deflection using filter](image)

![Figure 8. Frequency response of trailing-edge accelerometer](image)

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 simulations results prove that a fixed GPC algorithm with input constraints can provide flutter suppression for a wide range of flight conditions. The robustness properties of the GPC algorithm are also confirmed with the wind-tunnel test.

**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 the desirable performance characteristics. Test results without frequency weighting and control inputs constrained to ±3 degrees are given in Figures 9 and 10. The curve in Figure 9 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 Figure 10, the trailing-edge acceleration was maintained about zero throughout the test.

There are several desirable control characteristics that need to be incorporated in the design of an active flutter suppression system. The closed-loop system must be able to modeling errors since flutter is a highly nonlinear phenomena. The controller must also be able to dampen the flutter to some acceptable magnitude, within an allowable time period, and with minimal control surface deflection. In the case of the BACT plant, the damping time is not needed to be as short as with a high performance wing. The commanded inputs were only used for flutter suppression. If they were also used for flight control the input constraints would need to be much smaller.

Conclusion

The results from the wind-tunnel test showed that for the tested flight conditions, 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 GPC two improvements are being considered. First, a nonlinear neural network model of the BACT plant is being developed. This would allow 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 would be to use more than one accelerometer and control surface to dampen the flutter. These improvements could lead to a shorter damping time with less control surface movement and increase the capability of GPC to control more difficult flight conditions.

References