Real-Time Global Nonlinear Aerodynamic Modeling for Learn-To-Fly

Eugene A. Morelli[1]
NASA Langley Research Center, Hampton, Virginia, 23681

Flight testing and modeling techniques were developed to accurately identify global nonlinear aerodynamic models for aircraft in real time. The techniques were developed and demonstrated during flight testing of a remotely-piloted subscale propeller-driven fixed-wing aircraft using flight test maneuvers designed to simulate a Learn-To-Fly scenario. Prediction testing was used to evaluate the quality of the global models identified in real time. The real-time global nonlinear aerodynamic modeling algorithm will be integrated and further tested with learning adaptive control and guidance for NASA Learn-To-Fly concept flight demonstrations.

Nomenclature

a_x, a_y, a_z	=	body-axis translational accelerometer measurements, g
b	=	wing span, ft
$\bar{c}$	=	wing mean aerodynamic chord, ft
C_X, C_Y, C_Z	=	body-axis nondimensional aerodynamic force coefficients
C_l, C_m, C_n	=	body-axis nondimensional aerodynamic moment coefficients
$E\{\cdot\}$	=	expected value
$I_{xx}, I_{yy}, I_{zz}, I_{xz}$	=	inertia tensor elements, slug-ft^2
m	=	mass, slug
N	=	number of data points
p, q, r	=	body-axis roll, pitch, and yaw rates, rad/s or deg/s
$\bar{q}$	=	dynamic pressure, lbf/ft^2
s	=	standard error
S	=	wing reference area, ft^2
T	=	x body-axis component of engine thrust, lbf
V	=	true airspeed, ft/s
α	=	angle of attack, rad or deg
β	=	sideslip angle, rad or deg
$\delta_a, \delta_e, \delta_r, \delta_{rv}$	=	aileron, elevator, rudder, and ruddervator deflections, rad or deg
σ^2	=	fit error variance
ϕ, θ, ψ	=	Euler roll, pitch, and yaw angles, rad or deg
$\boldsymbol{\Sigma}$	=	covariance matrix

Subscripts

cg	=	center of gravity
L	=	left
o	=	reference value or base term
p	=	prior
R	=	right

[1] Research Engineer, Dynamic Systems and Control Branch, MS 308, AIAA Associate Fellow

Superscripts

T	=	transpose
$\hat{}$	=	estimate
$\dot{}$	=	time derivative
-1	=	matrix inverse
$\bar{}$	=	mean

I. Introduction

THE NASA Learn-To-Fly initiative seeks to autonomously develop a global aircraft model and full-envelope flight control in real time, with minimal ground testing, human interaction, and analysis time. This represents a new paradigm for developing and flight testing new or modified aircraft, as depicted in Fig. 1. The new paradigm replaces conventional ground-based testing and analysis with real-time methods applied in flight. The main payoffs are vastly improved efficiency in time and money, and the development of rapid adaptive processes that are generally applicable and globally valid. The Learn-To-Fly concept is an enabling technology for rapid aircraft prototyping and testing, but also has applications in the areas of fault detection, self-learning vehicles, flight envelope protection, safe and reliable flight operations for unmanned air vehicles, and rapidly generating or updating aerodynamic models for flight simulation, among others.

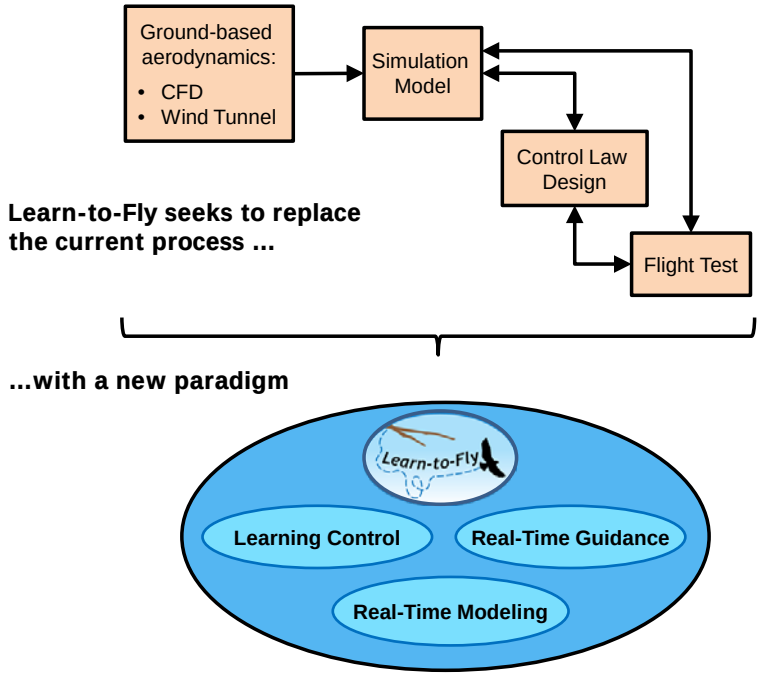

Figure 1. **Comparison of conventional aircraft development with the Learn-To-Fly concept**

A key component of the Learn-To-Fly concept is real-time global nonlinear aerodynamic modeling based on flight data alone. The conventional state-of-the-art for global aerodynamic modeling typically involves iterative ground-based testing and analysis, numerous precisely-flown maneuvers, and significant efforts in post-flight analysis[1-5], as shown in the upper part of Fig. 1. This approach is expensive and time-consuming, and the ground-based tools have inherent fidelity limitations arising from factors such as wind-tunnel model scale and geometry differences relative to the full-scale aircraft, wind-tunnel wall and sting interference, wind-tunnel flow angularity, Reynolds number differences, flow modeling deficiencies, and grid geometry approximations for both the aircraft and the flow field. Using flight test methods to generate a global aerodynamic model directly avoids all of these problems, although the typical problems associated with any flight test still remain, e.g., sensor data quality, achieving adequate data information content from the flight test maneuvers, flight test risk and expense, and

practical constraints. Recent flight research[6-10] has demonstrated that novel efficient flight test maneuvers (both automated and implemented by a pilot) can be combined with advanced nonlinear modeling techniques to achieve accurate global nonlinear aerodynamic models in near real time for all 6 rigid-body degrees of freedom simultaneously, based on flight data alone.

The goal of the current research is to further develop these flight test and modeling techniques to achieve real-time autonomous operation, and to examine practical issues involved in the flight testing and real-time modeling envisioned for the Learn-To-Fly initiative. The investigations described here include examining the data information content from a typical Learn-To-Fly flight experiment, investigating timing and practical issues associated with onboard autonomous software implementation, and evaluating the quality of the identified real-time models. The global aerodynamic modeling method developed in this work executes autonomously in real time with reasonable computational requirements, using flight data alone, and produces accurate global aerodynamic models with good prediction capability.

The next section describes the test aircraft. Section III contains a description of the flight test maneuvers, with some discussion and evaluation of the flight data. Section IV describes the real-time global aerodynamic modeling technique applied in flight, based on multivariate orthogonal function modeling[7,9-12] and a recursive QR decomposition[13-15]. Results of the autonomous real-time global nonlinear modeling from flight data and associated prediction testing appear in Section V, and conclusions are presented in Section VI.

All of the real-time software used in this work was written in MATLAB®. Some of the software used for experiment design, real-time data analysis, and real-time multivariate orthogonal function modeling tasks came from the software toolbox called System IDentification Programs for AirCraft, or SIDPAC[12,16].

II. Bat-4 Aircraft

The test aircraft was a Bat-4 remotely-piloted airplane, shown in Fig. 2. The Bat-4 is a commercially-available fixed-wing airplane powered by a 110 cc gasoline engine driving a fixed-pitch pusher propeller. Cruising speed is approximately 50 kts, and maximum speed is 70 kts. The conventional tricycle landing gear is fixed and ruggedized. The test aircraft was equipped with pitot tubes and angle-of-attack and sideslip angle vanes mounted on wingtip booms. Control surfaces are conventional ailerons and trailing-edge flaps on the wings, along with ruddervators on each of two anhedral tail surfaces. Aircraft geometry and nominal mass properties are given in Table 1.

The control surface deflections were not measured directly, but instead were estimated using the control surface commands and an actuator model identified from flight data for the same actuators flown on a different aircraft[6,7]. The actuator model was a first-order dynamic system with break frequency at 5 Hz and a pure time delay equal to 10 ms. This actuator model can be expressed in transfer function form as

$$\delta = \frac{31.4}{(s+31.4)} e^{-0.01s} \delta_c \tag{1a}$$

where δ is the control surface deflection and δ_c is the corresponding command. The control surface deflections were calculated in real time using command data and the actuator model.

For the flight data analyzed in this work, only the ruddervators and ailerons were deflected. Symmetric ruddervator deflections were used for pitch control, and asymmetric ruddervator deflections were used for yaw control. Left and right ailerons were deflected asymmetrically, in the conventional way, for roll control. Definitions of the effective control surface deflections are given below. Trailing edge down is positive deflection for all control surfaces.

$$\delta_e = \frac{1}{2}\left(\delta_{rv_R} + \delta_{rv_L}\right) \qquad \delta_r = \frac{1}{2}\left(\delta_{rv_R} - \delta_{rv_L}\right) \qquad \delta_a = \frac{1}{2}\left(\delta_{a_R} - \delta_{a_L}\right) \tag{1b}$$

A research pilot executed the flight test maneuvers from inside a mobile control room, using a synthetic vision display drawn from telemetry data and a local terrain database, along with video from a camera in the nose of the aircraft. Inputs from the research pilot and a ground-based flight control system were used to generate control surface commands which were transmitted via telemetry to the aircraft. Control surface deflections were computed from command data telemetered to the ground along with the aircraft response data, to enforce uniform telemetry latency in the flight data.

The flight control system has the capability to inject automated control surface perturbations to excite the aircraft dynamic response for modeling purposes. These control surface perturbations can have arbitrary waveforms, and can be applied to multiple control surfaces individually or simultaneously. The perturbations are summed with pilot and feedback control commands in the flight control system, just before the actuator command rate and position limiting. The research pilot flies the aircraft to the desired flight condition(s), then activates the automated control surface perturbations with a button on the throttle control.

Figure 2. **Bat-4 aircraft**
(Credit : NASA Wallops Flight Facility)

The Bat-4 aircraft was equipped with a micro-INS, which provided 3-axis translational accelerometer measurements, angular rate measurements, estimated attitude angles, and GPS position and velocity. Air data probes attached to booms mounted on each wingtip (visible in Fig. 2) measured angle of attack, sideslip angle, static pressure, and dynamic pressure. Measurements from static pressure sensors and ambient temperature sensors were used to compute air density and altitude. Propeller speed was measured in rpm. Mass properties were computed based on measured fuel flow, pre-flight weight and balance, and inertia estimates made from a detailed 3D computer model of the aircraft and payload. Pilot stick and rudder pedal commands and throttle position were also measured and recorded. Flight data were telemetered to the ground in real time. Sampling rate for the flight data was 200 Hz, downsampled to 25 Hz for real-time data analysis and modeling.

Table 1. **Bat-4 aircraft geometry and nominal mass properties**

$\bar{c}$, ft	1.50
b , ft	12.68
S , ft^2	19.02
x_{cg} , in	48.98
y_{cg} , in	0.036
z_{cg} , in	24.01
m , slugs	3.593
I_{xx} , slugs-ft^2	9.841
I_{yy} , slugs-ft^2	8.861
I_{zz} , slugs-ft^2	16.17
I_{xz} , slugs-ft^2	-1.015

III. Flight Test Data

To enable accurate identification of the aircraft global aerodynamics, flight data must cover a wide range for the explanatory variables and have sufficient information content for accurate model identification. Prior research[6,7] has demonstrated that applying automated orthogonal optimized multisine perturbation inputs to the control surfaces during slow transitions through a range of nominal flight conditions is an excellent method for collecting global aerodynamic modeling data with good information content.

A modified global maneuver of this general type was flown on the Bat-4. The maneuver was intended to simulate dropping an aircraft from altitude in a wings-level nose-up attitude, as might be done for a Learn-To-Fly scenario. Automated orthogonal optimized multisine excitations were applied to the control surfaces throughout this maneuver. The pilot flew the Bat-4 to a steady, slow, nose-high flight condition, then simultaneously released back pressure on the stick and held in the button on the throttle to implement the automated optimized multisine inputs in all axes. As the aircraft pitched forward to lower angles of attack, the pilot then slowly eased back on the stick until the aircraft again reached a slow speed, nose-high flight condition, then released back pressure on the stick, to repeat the maneuver. Automated excitation inputs were applied to the control surfaces continuously throughout the maneuver. Figure 3 shows flight data from four Learn-To-Fly maneuvers flown sequentially and continuously.

The Learn-To-Fly maneuver described here is a very high-efficiency global maneuver that provides multi-axis flight data with high information content and low correlations among the explanatory variables over a large range of flight conditions, as required for accurate and efficient global aerodynamic modeling. Such maneuvers are practical because the energetic multisine perturbation inputs excite the aircraft dynamics rapidly and effectively in all axes simultaneously, as the flight condition is changing slowly.

Figure 4 shows cross plots of selected explanatory variable data from the Learn-To-Fly maneuvers shown in Fig. 3. These plots demonstrate that a wide range of these explanatory variables was swept through during the maneuvers. Note also that the cross plots generally do not resemble straight lines, which means that the explanatory variable data from this maneuver had very low pairwise correlations. Low correlations mean that the dependencies of the aircraft aerodynamics on the explanatory variables can be identified accurately and without ambiguity. Cross plots for other aircraft states and controls typically used for global aerodynamic modeling were similar to Fig. 4 in that the plots indicated low correlations and wide ranges of coverage for the explanatory variables.

Local maneuvers were also flown, where automated orthogonal optimized multisine inputs were applied to the control surfaces at a single nominal flight condition. Multi-axis flight data from these maneuvers were used for global model validation and prediction testing. Table 2 contains a list of the research flight test maneuvers from Bat-4 flight 3 that were used in this work for real-time global modeling and prediction.

Table 2. **Bat-4 Flight 3 Research Flight Data**

Maneuver	Maneuver Description and Objective
29a	Multi-axis orthogonal optimized multisine, nominal flight condition, low amplitude, input evaluation and local prediction
29b	Multi-axis orthogonal optimized multisine, nominal flight condition, medium amplitude, input evaluation and local prediction
29c	Multi-axis orthogonal optimized multisine, nominal flight condition, medium amplitude, input evaluation and local prediction
39a	Multi-axis orthogonal optimized multisine, Learn-To-Fly maneuver, global aerodynamic modeling
39b	Multi-axis orthogonal optimized multisine, Learn-To-Fly maneuver, global aerodynamic modeling
39c	Multi-axis orthogonal optimized multisine, Learn-To-Fly maneuver, global aerodynamic modeling
39d	Multi-axis orthogonal optimized multisine, Learn-To-Fly maneuver, global aerodynamic modeling

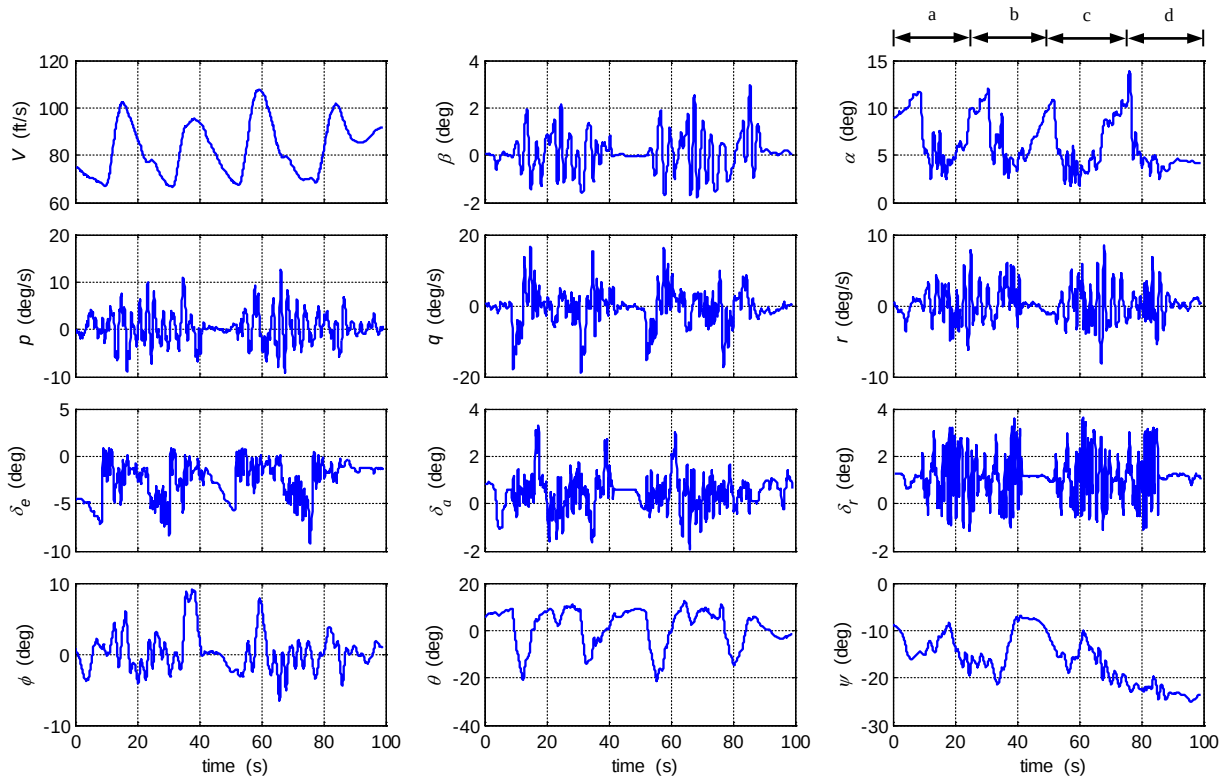

Figure 3. **Bat-4 flight data for Learn-To-Fly global aerodynamic modeling maneuvers**

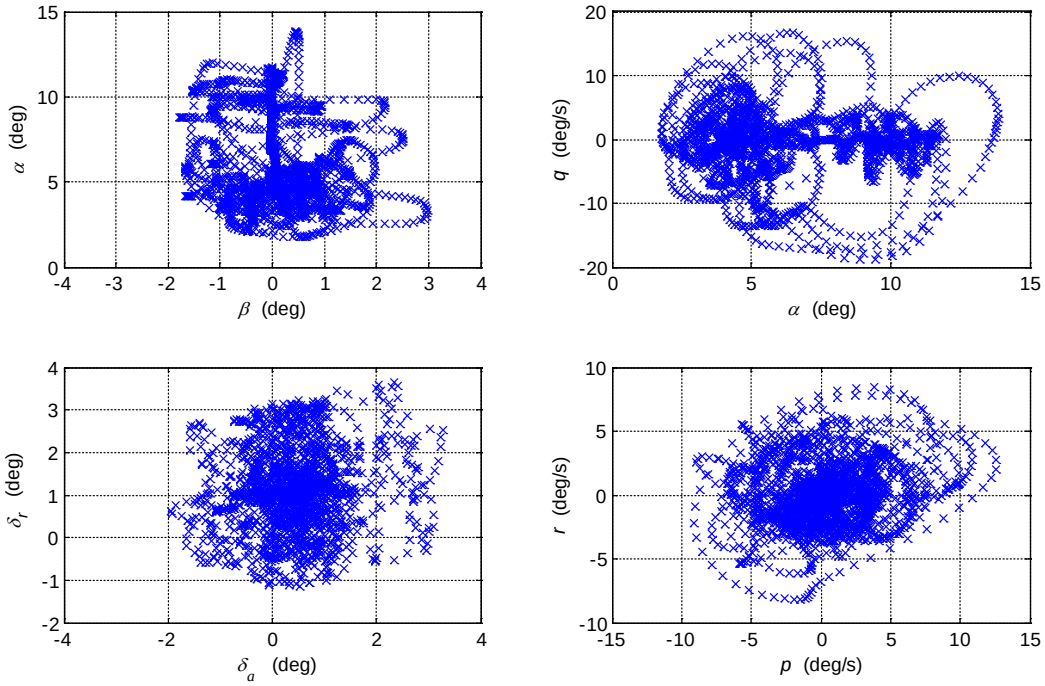

Figure 4. **Bat-4 flight data cross plots for Learn-To-Fly global aerodynamic modeling maneuvers**

IV. Real-Time Global Aerodynamic Modeling

The objective is to identify a global nonlinear model in real time for each nondimensional aerodynamic force and moment coefficient as a function of explanatory variables that can be measured, such as angle of attack, sideslip angle, body-axis angular rates, and control surface deflections. There are two main difficulties in doing this: 1) designing an experiment to collect informative dynamic data over large ranges of the explanatory variables, which was addressed in Section III, and 2) identifying an accurate global nonlinear model in real time, which is the subject of this section.

One approach to global aerodynamic modeling that has been successful in offline and quasi-real time applications is orthogonalizing candidate linear and nonlinear multivariate modeling functions to isolate and quantify the modeling capability of each individual model term, then applying a statistical metric computed from the data to select which of the terms should be included in the final model[7,9-12]. The multivariate orthogonal function modeling used in this work employs this general approach to identify compact global nonlinear models valid over large ranges of the explanatory variables. However, to use this modeling approach in real time, a method for recursively orthogonalizing arbitrary data vectors is required. Once that problem is solved, the same computational methods used successfully in the past for global nonlinear model identification can be applied in real time. The method for recursive orthogonalization of arbitrary data vectors (i.e., arbitrary multivariate modeling functions) is given later in this section.

For very large ranges of the explanatory variables, or severe local nonlinearities, using orthogonal multivariate polynomial functions for the modeling sometimes compromises local model fit in order to achieve a better global model fit. This can happen because orthogonal multivariate polynomial functions do not have the ability to change locally without modifying the entire model. The problem can be remedied by also including spline functions of the explanatory variables in the modeling process[7,10,17], which can provide the required local nonlinear modeling capability, while retaining clear physical interpretation of the model and the automated selection of optimal model complexity necessary to accurately characterize the functional dependencies. Considering spline functions (or any other multivariate function) as candidate model terms can be done recursively in the same way as any other arbitrary modeling function of the explanatory variables.

The next subsections describe the general nonlinear aerodynamic modeling problem using nondimensional aerodynamic coefficients, model quality metrics used to evaluate the success of the modeling, and the method developed for real-time global aerodynamic modeling.

A. Nondimensional Aerodynamic Coefficient Modeling

For global aerodynamic modeling from flight data, nondimensional aerodynamic force and moment coefficients were used as the response variables (also called the dependent variables) for the modeling problem. A separate modeling problem was solved for each force or moment coefficient, corresponding to minimizing the squared equation error in each individual equation of motion for the six rigid-body degrees of freedom of the aircraft[12]. Values for the nondimensional aerodynamic force and moment coefficients cannot be measured directly in flight, but instead must be computed from measured and known quantities using the following equations[12]

$$C_X = \frac{\left(m a_x - T\right)}{\bar{q} S} \qquad C_Y = \frac{m a_y}{\bar{q} S} \qquad C_Z = \frac{m a_z}{\bar{q} S} \tag{2}$$

$$C_l = \frac{I_{xx}}{\bar{q} S b}\left[\dot{p} - \frac{I_{xz}}{I_{xx}}\left(pq + \dot{r} \right) + \frac{\left(I_{zz} - I_{yy} \right)}{I_{xx}} qr \right] \tag{3}$$

$$C_m = \frac{I_{yy}}{\bar{q} S \bar{c}}\left[\dot{q} + \frac{\left(I_{xx} - I_{zz} \right)}{I_{yy}} pr + \frac{I_{xz}}{I_{yy}}\left(p^2 - r^2 \right) \right] \tag{4}$$

$$C_n = \frac{I_{zz}}{\bar{q} S b}\left[\dot{r} - \frac{I_{xz}}{I_{zz}}\left(\dot{p} - qr \right) + \frac{\left(I_{yy} - I_{xx} \right)}{I_{zz}} pq \right] \tag{5}$$

These expressions retain the full rigid-body nonlinear dynamics in the aircraft equations of motion.

The result is N values of the nondimensional force and moment coefficients, where N is the number of data points for a particular maneuver or series of maneuvers. Such data are often called measured force and moment coefficient data, even though the data are not measured directly, but rather computed from other measurements and known quantities. Data for explanatory variables such as angle of attack, sideslip angle, body-axis angular rates, and control surface deflections come from direct measurements.

Because angular accelerations $(\dot{p},\dot{q},\dot{r})$ were not measured, a smoothed local differentiation method[12,18] was applied to the measured angular rate data to compute angular accelerations in real time. The data smoothing required to keep noise levels low for the computed angular accelerations used both past and future samples of the angular rates relative to the data point being smoothed, causing a delay of two time samples in the real-time results. Using 25 Hz data, this amounted to an 0.08 s delay in the modeling results.

The desired form of the global aerodynamic model is a mathematical model structure with estimated model parameter values and associated uncertainties, relating the nondimensional aerodynamic force and moment coefficients to aircraft states and controls that can be measured. All of the global modeling was based on equation-error least-squares in the time domain. In this formulation, the dependent variable $\boldsymbol{z}$, which is one of the nondimensional force or moment coefficients, is modeled using an expansion of generally nonlinear model terms composed from the explanatory variables, which are nondimensional aircraft states and control surface deflections. This leads to an over-determined set of algebraic equations for the unknown model parameters, which can be solved using a standard least-squares method.

For example, the equation-error least-squares problem for the pitching moment coefficient C_m could be formulated using the model structure

$$C_m = C_{m_\alpha}\alpha + C_{m_q}\frac{q\bar{c}}{2V_o} + C_{m_{\delta_e}}\delta_e + C_{m_{\alpha\delta_e}}\alpha\delta_e + C_{m_o} \tag{6}$$

Equation (6) can be rewritten as

$$\boldsymbol{z} = \boldsymbol{X\theta} + \boldsymbol{\varepsilon} \tag{7}$$

where

$\boldsymbol{z} = \left[C_m(1) \;\; C_m(2) \;\; \ldots \;\; C_m(N)\right]^T = N \times 1$ vector of values computed from Eq. (4)

$\boldsymbol{\theta} = \left[C_{m_\alpha} \;\; C_{m_q} \;\; C_{m_{\delta_e}} \;\; C_{m_{\alpha\delta_e}} \;\; C_{m_o}\right]^T = 5 \times 1$ vector of unknown parameters

$\boldsymbol{X} = \left[\boldsymbol{\alpha} \;\; \frac{\boldsymbol{q\bar{c}}}{2V_o} \;\; \boldsymbol{\delta_e} \;\; \boldsymbol{\alpha\delta_e} \;\; \mathbf{1}\right] = N \times 5$ matrix of modeling function or regressor vectors

$\boldsymbol{\varepsilon} = \left[\varepsilon(1) \;\; \varepsilon(2) \;\; \ldots \;\; \varepsilon(N)\right]^T = N \times 1$ vector of equation errors

The matrix $\boldsymbol{X}$ is assembled using measured data, with each column representing a modeling function, also called a regressor. The best estimator of $\boldsymbol{\theta}$ in a least-squares sense comes from minimizing the sum of squared differences between the dependent variable measurements $\boldsymbol{z}$ and the model,

$$J(\boldsymbol{\theta}) = \frac{1}{2}(z - \boldsymbol{X\theta})^T(z - \boldsymbol{X\theta}) \tag{8}$$

The least-squares solution for the unknown parameter vector $\boldsymbol{\theta}$ is found by taking the derivative of the cost function in Eq. (8) with respect to $\boldsymbol{\theta}$, setting the result equal to zero, and solving for $\boldsymbol{\theta}$,[12]

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = -\boldsymbol{X}^T \boldsymbol{z} + \boldsymbol{X}^T \boldsymbol{X} \hat{\boldsymbol{\theta}} = 0 \tag{9a}$$

$$\boldsymbol{X}^T \boldsymbol{X} \hat{\boldsymbol{\theta}} = \boldsymbol{X}^T \boldsymbol{z} \tag{9b}$$

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{X}^T \boldsymbol{X} \right)^{-1} \boldsymbol{X}^T \boldsymbol{z} \tag{9c}$$

The estimated parameter covariance matrix is computed from[12]

$$\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\theta}}\right) \equiv E\left[\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)^T \right] = \hat{\sigma}^2 \left(\boldsymbol{X}^T \boldsymbol{X} \right)^{-1} \equiv \left[\Sigma_{ij} \right] \qquad i, j = 1,2,\ldots,n \tag{10}$$

The model output is

$$\hat{\boldsymbol{y}} = \boldsymbol{X} \hat{\boldsymbol{\theta}} \tag{11}$$

and the model fit error variance estimate is

$$\hat{\sigma}^2 = \frac{\left(\boldsymbol{z} - \hat{\boldsymbol{y}}\right)^T \left(\boldsymbol{z} - \hat{\boldsymbol{y}}\right)}{\left(N - n\right)} \tag{12}$$

where n is the number of unknown parameters, so that $n = 5$ for this example. The standard errors of the estimated parameters are given by the square root of the diagonal elements of the covariance matrix,

$$s\left(\hat{\theta}_j\right) = \sqrt{\Sigma_{jj}} \qquad j = 1,2,\ldots,n \tag{13}$$

This approach allows modeling functions that are nonlinear in the explanatory variables (such as $\alpha\delta_e$ in Eq. (6)), but the solution in Eqs. (9)-(13) involves non-iterative linear algebra, which can be done very rapidly. This is because the model outputs are linearly related to the unknown parameters, cf. Eq. (11), which means the least-squares cost function is quadratic in the unknown parameters. For global aerodynamic modeling, a main issue is identifying which modeling functions should be used for the aerodynamic coefficients, or equivalently, determining which functions should appear on the right side of model equations like Eq. (6). This important issue is addressed in subsequent sections describing multivariate orthogonal function modeling.

B. Modeling Metrics

Two modeling metrics were used to quantify the success of the global aerodynamic modeling. The modeling metrics were:

1. Coefficient of determination R^2

$$R^2 = \frac{\sum_{i=1}^{N} \left[\hat{y}(i) - \bar{\boldsymbol{z}} \right]^2}{\sum_{i=1}^{N} \left[z(i) - \bar{\boldsymbol{z}} \right]^2} = 1 - \frac{\sum_{i=1}^{N} \left[z(i) - \hat{y}(i) \right]^2}{\sum_{i=1}^{N} \left[z(i) - \bar{\boldsymbol{z}} \right]^2} = 1 - \frac{\left(\boldsymbol{z} - \hat{\boldsymbol{y}}\right)^T \left(\boldsymbol{z} - \hat{\boldsymbol{y}}\right)}{\boldsymbol{z}^T \boldsymbol{z} - N \bar{\boldsymbol{z}}^2} \tag{14}$$

The R^2 metric quantifies the fraction of the variation in the dependent variable about its mean value that is explained by the model, so that $0 \le R^2 \le 1$. Often, R^2 is given as a percentage. R^2 is a model fit quality measure.

2. Predicted Squared Error PSE

$$PSE = \frac{1}{N}\left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\theta}}\right)^T \left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\theta}}\right) + \sigma_{max}^2 \frac{n}{N} \tag{15}$$

or

$$PSE = \frac{2}{N} J\left(\hat{\boldsymbol{\theta}}\right) + \sigma_{max}^2 \frac{n}{N} \tag{16}$$

The *PSE* metric quantifies expected squared prediction error for an identified model when applied to data not used in the model identification process. The constant σ_{max}^2 is an upper-bound estimate of the squared error between future data and the model, i.e., the upper-bound squared error for prediction cases. A simple estimate of σ_{max}^2 that is independent of the model structure can be obtained in real time by computing a running average of squared high-pass filtered dependent variable data. This simple quantification of the dependent variable noise variance can then be multiplied by a factor to implement a conservative estimate for σ_{max}^2. The process is analogous to choosing a confidence level for an *F*-ratio test in model structure determination, as will become clear in the later discussion on model structure determination. In this work, a second-order Butterworth high-pass filter with break frequency set at 2 Hz was applied to the dependent variable data, and the result was squared and averaged in real time to obtain a real-time noise variance estimate. That value was multiplied by a factor of 25 to obtain a conservative estimate of σ_{max}^2,

$$\begin{aligned} \hat{\sigma}_i^2 &= \left[\hat{\sigma}_{i-1}^2 (i-1) + v_i^2\right] / i \\ \hat{\sigma}_{max}^2 &= 25\hat{\sigma}^2 \quad \text{or} \quad \hat{\sigma}_{max} = 5\hat{\sigma} \end{aligned} \tag{17}$$

where v_i is the high-pass filtered dependent variable data at the *i*th time step. Using a conservative upper bound estimate for σ_{max}^2 means the *PSE* metric will tend to overestimate actual squared prediction errors for new data. Therefore, the *PSE* metric conservatively estimates the squared error for prediction cases. Various high-pass filter designs and break frequencies could be used as well. The objective of the high-pass filter design is to isolate components of the dependent variable data that can be used to characterize the magnitude of the noise for model structure determination. The *PSE* metric is a combination of a model fit quality measure (the first term on the right side of Eq. (16), proportional to the least-squares cost function) and a model complexity penalty (the second term on the right side of Eq. (16), proportional to the number of terms in the model, *n*).

C. Multivariate Orthogonal Function Modeling

The multivariate orthogonal function modeling was based on previous work[7,9-12], with modifications to achieve real-time operation. The technique begins by generating candidate multivariate functions of the selected explanatory variable data, up to a selected maximum model complexity. Although any function of the explanatory variables could be used, multivariate polynomials and spline functions are preferred because of their similarity to a truncated Taylor series and their easy physical interpretation. These ordinary functions are then orthogonalized, so that each of the resulting orthogonal functions retains only the explanatory capability that is unique to that modeling function. With this data transformation, it is a straightforward process to select which of the orthogonal modeling functions are most effective in modeling the measured data for the dependent variable, and how many of these orthogonal functions should be included to identify a model that exhibits both a good fit to the modeling data and good prediction capability for other data. The final steps are an error-free transformation from the selected orthogonal modeling functions back to physically-meaningful ordinary functions of the explanatory variables, and calculation of the uncertainties for the associated model parameter estimates.

1. Generating Orthogonal Modeling Functions Recursively

In previous work[7,9-12], multivariate orthogonal functions were generated from ordinary multivariate functions in the explanatory variables using a Gram-Schmidt orthogonalization procedure applied to all of the data at once. This

approach is not efficient enough for real-time operation. Instead, recursive orthogonalization can be implemented using a standard $\boldsymbol{QR}$ decomposition of the matrix of candidate regressors,

$$\boldsymbol{X} = \boldsymbol{QR} \tag{18}$$

where the columns of $\boldsymbol{X}$ contain the candidate modeling functions, $\boldsymbol{Q}$ is an orthonormal matrix with the same dimensions as $\boldsymbol{X}$, and $\boldsymbol{R}$ is a square upper triangular matrix. The recursive $\boldsymbol{QR}$ decomposition process is initialized by applying a $\boldsymbol{QR}$ decomposition algorithm to the $\boldsymbol{X}$ matrix built from the first n_c data points, where n_c is the number of candidate modeling functions. Implementations of $\boldsymbol{QR}$ decomposition algorithms are available in many numerical analysis software packages, including MATLAB®, which was used for this work. Substituting the decomposition in Eq. (18) into Eq. (9b),

$$\boldsymbol{R}^T\boldsymbol{R}\hat{\boldsymbol{\theta}} = \boldsymbol{R}^T\boldsymbol{Q}^T\boldsymbol{z} \tag{19}$$

where $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$ for the orthonormal matrix $\boldsymbol{Q}$. Assuming $\boldsymbol{R}$ is nonsingular,

$$\boldsymbol{R}\hat{\boldsymbol{\theta}} = \boldsymbol{Q}^T\boldsymbol{z} \tag{20}$$

From Eq. (20), the elements of $\hat{\boldsymbol{\theta}}$ can be found by simple back substitution, because $\boldsymbol{R}$ is an upper triangular matrix. Note that Eq. (20) is just an alternate form of Eq. (9b). However, Eq. (20) is convenient for recursion, because the $\boldsymbol{R}$ matrix must be an upper diagonal $n_c \times n_c$ matrix, and only the inner products of the orthonormal columns of $\boldsymbol{Q}$ with the dependent variable vector $\boldsymbol{z}$ appear in the equation, and not the $\boldsymbol{Q}$ matrix itself. Consequently, the dimension of both sides of Eq. (20) is always $n_c \times 1$, regardless of the number of data points *N*.

Re-writing Eq. (20) in component form,

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n_c} \\ 0 & r_{22} & \cdots & r_{2n_c} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & r_{n_cn_c} \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_{n_c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1^T\boldsymbol{z} \\ \boldsymbol{q}_2^T\boldsymbol{z} \\ \vdots \\ \boldsymbol{q}_{n_c}^T\boldsymbol{z} \end{bmatrix} \tag{21}$$

where $\boldsymbol{q}_j$, is the *j*th column of the matrix $\boldsymbol{Q}$. The right side of Eq. (21) is a vector of projections of the dependent variable vector $\boldsymbol{z}$ onto the orthonormal functions in the columns of $\boldsymbol{Q}$. The absolute values of these quantities indicate the degree of correlation of the orthonormal functions in the columns of $\boldsymbol{Q}$ with $\boldsymbol{z}$, and consequently, the effectiveness of each orthonormal function in modeling the dependent variable data.

When new data arrive, Eq. (21) is augmented by appending the new data in the bottom row,

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n_c} \\ 0 & r_{22} & \cdots & r_{2n_c} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & r_{n_cn_c} \\ \xi_1 & \xi_2 & \cdots & \xi_{n_c} \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_{n_c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1^T\boldsymbol{z} \\ \boldsymbol{q}_2^T\boldsymbol{z} \\ \vdots \\ \boldsymbol{q}_{n_c}^T\boldsymbol{z} \\ \zeta \end{bmatrix} \tag{22}$$

where $\begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_{n_c} \end{bmatrix}$ is the new row of data for the $\boldsymbol{X}$ matrix, and ζ is the new dependent variable data. To maintain the $\boldsymbol{QR}$ decomposition including the appended data, the matrix multiplying the parameter vector must be

transformed so that the last row contains all zeros. This can be done by applying Givens rotation matrices[13-14]. For example, to remove the value of ξ_1 from the bottom row, the Givens rotation matrix is

$$\boldsymbol{G}_1 = \begin{bmatrix} c & 0 & \dots & 0 & s \\ 0 & 1 & \dots & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & 0 \\ -s & 0 & \dots & 0 & c \end{bmatrix} \tag{23a}$$

where

$$c = r_{11} \Big/ \sqrt{r_{11}^2 + \xi_1^2} \qquad s = \xi_1 \Big/ \sqrt{r_{11}^2 + \xi_1^2} \tag{23b}$$

Applying the rotation matrix $\boldsymbol{G}_1$ to Eq. (22) gives

$$\begin{bmatrix} cr_{11} + s\xi_1 & cr_{12} + s\xi_2 & \dots & cr_{1n_c} + s\xi_{n_c} \\ 0 & r_{22} & \dots & r_{2n_c} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & r_{n_c n_c} \\ 0 & -sr_{12} + c\xi_2 & \dots & -sr_{1n_c} + c\xi_{n_c} \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_{n_c} \end{bmatrix} = \boldsymbol{G}_1 \begin{bmatrix} \boldsymbol{q}_1^T \boldsymbol{z} \\ \boldsymbol{q}_2^T \boldsymbol{z} \\ \vdots \\ \boldsymbol{q}_{n_c}^T \boldsymbol{z} \\ \zeta \end{bmatrix} \tag{24}$$

which makes the lower left element in the matrix equal to zero. This process is repeated until the last row of the matrix on the left side contains all zeros, which indicates that the $\boldsymbol{QR}$ decomposition has been updated. In general, this will involve n_c rotation matrices, which are applied to both sides of Eq. (22), resulting in

$$\begin{bmatrix} r'_{11} & r'_{12} & \dots & r'_{1n_c} \\ 0 & r'_{22} & \dots & r'_{2n_c} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & r'_{n_c n_c} \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_{n_c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1'^T \boldsymbol{z} \\ \boldsymbol{q}_2'^T \boldsymbol{z} \\ \vdots \\ \boldsymbol{q}_{n_c}'^T \boldsymbol{z} \\ \varepsilon \end{bmatrix} \tag{25}$$

where the primed notation indicates the updated $\boldsymbol{QR}$ decomposition, including the appended data. The value ε remaining in the bottom row on the right side of Eq. (25) is the residual for the appended data point, assuming a model that includes all the $\boldsymbol{q}_j$, $j = 1, 2, \dots, n_c$. Equivalently, this is the portion of the appended dependent variable data that cannot be projected onto the updated orthonormalized candidate modeling functions. The combined rotation matrix for the orthonormalization update process is

$$\boldsymbol{G} = \boldsymbol{G}_{n_c} \dots \boldsymbol{G}_2 \boldsymbol{G}_1 \tag{26}$$

After all of the rotations have been applied to arrive at Eq. (25), the bottom row of zeros in the matrix on the left side is discarded. The residual ε is squared and added to a running sum of squared residuals, then discarded. The entire process is then repeated as each new data sample arrives. The result is a recursive algorithm that efficiently updates the orthonormalization of the n_c candidate modeling functions using simple matrix multiplications after each new data point is appended to the result from the previous time step. The recursive orthonormalization is

implemented with n_c simple matrix multiplications, which can be done very efficiently. Note that the model parameters $\hat{\theta}_j$, $j = 1, 2, \ldots, n_c$, are associated with the original multivariate modeling functions in the columns of $\boldsymbol{X}$, and not with the orthonormal functions in the columns of $\boldsymbol{Q}$.

2. Least Squares Parameter Estimation using Orthonormal Functions

The form of a multivariate orthonormal function model is

$$\boldsymbol{z} = a_1 \boldsymbol{q}_1 + a_2 \boldsymbol{q}_2 + \ldots + a_n \boldsymbol{q}_n + \boldsymbol{\varepsilon} \tag{27}$$

where $\boldsymbol{z}$ is an *N*-dimensional vector of dependent variable data (e.g., nondimensional force or moment coefficient data), $\boldsymbol{z} = \left[z_1, z_2, \ldots, z_N \right]^T$, modeled in terms of a linear combination of *n* mutually orthonormal modeling functions $\boldsymbol{q}_j$, $j = 1, 2, \ldots, n$. Each $\boldsymbol{q}_j$ is an *N*-dimensional vector that in general depends on the explanatory variables. The a_j, $j = 1, 2, \ldots, n$ are constant model parameters to be determined, and $\boldsymbol{\varepsilon}$ denotes the modeling error vector.

Equation (27) represents a mathematical model for the dependent variable $\boldsymbol{z}$ in terms of orthonormal functions generated from the explanatory variable data. The important question of determining which modeling functions should be included in Eq. (27), which implicitly determines n, will be addressed now.

Assembling the *n* orthonormal modeling functions from Eq. (27) in the columns of an $N \times n$ matrix $\boldsymbol{Q}$,

$$\boldsymbol{Q} = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n \right] \tag{28}$$

and defining the unknown parameter vector $\boldsymbol{a} = \left[a_1, a_2, \ldots, a_n \right]^T$, Eq. (27) can be written as

$$\boldsymbol{z} = \boldsymbol{Q}\boldsymbol{a} + \boldsymbol{\varepsilon} \tag{29}$$

Equation (29) is the same model equation discussed earlier (cf. Eq. (7)), except that the modeling functions are now orthonormal functions. In this case, it is easier to determine an appropriate model structure, because the explanatory capability of each modeling function is completely distinct from all the others, due to the mutual orthogonality of the columns of $\boldsymbol{Q}$. This decouples the least squares modeling problem, as will be shown now.

For mutually orthonormal functions,

$$\boldsymbol{q}_i^T \boldsymbol{q}_j = \begin{cases} 1 \text{ for } i = j \\ 0 \text{ for } i \neq j \end{cases} \qquad i, j = 1, 2, \ldots, n \tag{30}$$

and $\boldsymbol{Q}^T\boldsymbol{Q}$ is the identity matrix. Using Eq. (30) in the least-squares solution from Eq.(9c), the *j*th element of the estimated parameter vector $\hat{\boldsymbol{a}}$ is given by

$$\hat{a}_j = \boldsymbol{q}_j^T \boldsymbol{z} \tag{31}$$

The least-squares cost function using orthonormal functions is then

$$J\left(\hat{\boldsymbol{a}}\right) = \frac{1}{2}\left[\boldsymbol{z}^T\boldsymbol{z} - \hat{\boldsymbol{a}}^T\hat{\boldsymbol{a}} \right] = \frac{1}{2}\left[\boldsymbol{z}^T\boldsymbol{z} - \sum_{j=1}^{n} \left(\boldsymbol{q}_j^T \boldsymbol{z} \right)^2 \right] \tag{32}$$

Equation (32) shows that when the modeling functions are orthonormal, the reduction in the least-squares cost function resulting from including the term $a_j \boldsymbol{q}_j$ in the model depends only on the dependent variable data $\boldsymbol{z}$ and the added orthonormal modeling function $\boldsymbol{q}_j$. The least-squares modeling problem is therefore decoupled, which means

each orthonormal modeling function can be evaluated independently in terms of its ability to reduce the least-squares model fit to the data, regardless of which other orthonormal modeling functions are already selected for the model. If the modeling functions were instead polynomials in the explanatory variables (or any other non-orthogonal function set), then the least squares problem would be coupled and iterative analysis would be required to find a subset of the candidate modeling functions for an adequate model structure.

The vector resulting from the recursive $\boldsymbol{QR}$ decomposition in Eq. (21) contains exactly the quantities $\boldsymbol{q}_j^T \boldsymbol{z}$ appearing in Eqs. (31) and (32). These quantities are calculated for all n_c candidate modeling functions, and are used to identify the model structure, which involves selecting the functions to be included in the model from the pool of n_c candidate modeling functions.

3. Model Structure Determination using Orthonormal Functions

The orthonormal modeling functions to be included in the model are chosen to minimize the predicted squared error metric *PSE* defined in Eq. (15)

$$PSE = \frac{1}{N}\left(\boldsymbol{z}-\boldsymbol{Q}\hat{\boldsymbol{a}}\right)^T\left(\boldsymbol{z}-\boldsymbol{Q}\hat{\boldsymbol{a}}\right)+\sigma_{max}^2\frac{n}{N} \tag{33}$$

Combining Eq. (33) with Eqs. (16) and (32),

$$PSE = \frac{1}{N}\left[\boldsymbol{z}^T\boldsymbol{z}-\sum_{j=1}^{n}\left(\boldsymbol{q}_j^T\boldsymbol{z}\right)^2\right]+\sigma_{max}^2\frac{n}{N} \tag{34}$$

The *PSE* depends on the mean squared fit error (*MSFE*),

$$MSFE = \frac{1}{N}\left(\boldsymbol{z}-\boldsymbol{Q}\hat{\boldsymbol{a}}\right)^T\left(\boldsymbol{z}-\boldsymbol{Q}\hat{\boldsymbol{a}}\right) = \frac{1}{N}\left[\boldsymbol{z}^T\boldsymbol{z}-\sum_{j=1}^{n}\left(\boldsymbol{q}_j^T\boldsymbol{z}\right)^2\right] \tag{35}$$

along with a term proportional to the number of terms in the model, n. The latter term prevents overfitting the data with too many model terms, which is detrimental to model prediction accuracy[12,19]. While the mean squared fit error must decrease with the addition of each orthonormal modeling function to the model (because $-\left(\boldsymbol{q}_j^T\boldsymbol{z}\right)^2$ is always negative), the overfit penalty term $\sigma_{max}^2 n/N$ must increase with each added model term (*n* increases). Introducing the orthonormal modeling functions into the model in order of most effective to least effective in reducing the mean squared fit error (quantified by $\left(\boldsymbol{q}_j^T\boldsymbol{z}\right)^2$ for the *j*th orthonormal modeling function) means that the *PSE* metric will always have a single global minimum. The recursive $\boldsymbol{QR}$ decomposition described earlier computes $\boldsymbol{q}_j^T\boldsymbol{z}$ for each of the orthonormalized candidate modeling functions $\boldsymbol{q}_j$, $j=1,2,...,n_c$.

Because the quantities $\boldsymbol{z}^T\boldsymbol{z}$, σ_{max}^2, and *N* depend only on the dependent variable data and therefore cannot be altered by the model, Eq. (34) shows that the criterion for including each $\boldsymbol{q}_j$ in the model can be reduced to

$$\left(\boldsymbol{q}_j^T\boldsymbol{z}\right)^2 > \sigma_{max}^2 \tag{36}$$

The criterion in Eq. (36) is the mathematical statement of a simple physical idea that only modeling functions that reduce the mean squared fit error by an amount that exceeds the maximum expected noise variance should be included in the model. This is the condition necessary for *PSE* to decrease when $\boldsymbol{q}_j$ is added to the model. Reference [19] contains further statistical arguments and analysis for the *PSE* metric, including justification for its use in modeling problems.

Using orthonormal functions to model the dependent variable data makes it possible to evaluate the merit of including each modeling function *individually*, based on the *PSE* metric. The goal is to select a model structure with minimum *PSE*, and the *PSE* always has a single global minimum for orthonormal modeling functions. Model structure determination based on the *PSE* metric is therefore a well-defined and straightforward process that can be (and was) automated.

In addition to the *PSE* criterion, each selected orthonormal function $\boldsymbol{q}_j$ was required to model at least a selected fraction of the total variation about the mean for the dependent variable, quantified by the R^2 metric,

$$\frac{\left(\boldsymbol{q}_j^T \boldsymbol{z}\right)^2}{\left(\boldsymbol{z}^T \boldsymbol{z} - N\bar{z}\right)} \geq \Delta R^2_{min} \tag{37}$$

which can be computed from the results of the recursive orthonormalization and simple real-time calculations using the dependent variable data,

$$\bar{z}_i = \left[\bar{z}_{i-1}(i-1) + z_i\right] / i \tag{38a}$$

$$\left(\boldsymbol{z}^T \boldsymbol{z}\right)_i = \left(\boldsymbol{z}^T \boldsymbol{z}\right)_{i-1} + z_i^2 \tag{38b}$$

For this work, ΔR^2_{min} was selected as 0.005, which corresponds to requiring each selected orthonormal function to model at least 0.5 percent of the total variation in the dependent variable about the mean.

The model parameters associated with the original modeling functions in the columns of the $\boldsymbol{X}$ matrix are determined from Eq. (20), using all rows and columns of the $\boldsymbol{R}$ matrix up to and including the index associated with the last element of vector $\boldsymbol{Q}^T\boldsymbol{z}$ selected for the model. Because the orthonormalization of the multivariate functions in the $\boldsymbol{X}$ matrix is sequential according to the order of the columns in $\boldsymbol{X}$, all prior rows and columns of $\boldsymbol{R}\hat{\boldsymbol{\theta}}$ must be included in the calculation of the model parameter estimates for each selected $\boldsymbol{q}_j$. Accordingly, if the last element of vector $\boldsymbol{Q}^T\boldsymbol{z}$ selected for the model is element *m*, only the elements $\hat{\theta}_j$, $j = 1, 2, \ldots, m$ are estimated – the remaining elements of $\hat{\boldsymbol{\theta}}$ are associated with candidate modeling functions that were not selected for the model, and were not involved in the orthonormalization of the selected orthonormal functions, so those parameters are zero.

The identified model output can be computed from

$$\boldsymbol{y} = \boldsymbol{X}_m \hat{\boldsymbol{\theta}}_m \tag{39}$$

where $\boldsymbol{X}_m$ and $\hat{\boldsymbol{\theta}}_m$ include only the first *m* columns of $\boldsymbol{X}$ and elements of $\hat{\boldsymbol{\theta}}$, respectively, and *m* is highest value of *j* for the orthonormal modeling functions selected for the model, or equivalently, the number of the last $\boldsymbol{Q}^T\boldsymbol{z}$ vector element selected for the model. Note that

$$n \leq m \leq n_c \tag{40}$$

because the number of orthonormal functions selected for the model, *n*, can be less than the highest index for selected orthonormal modeling functions, *m*, and both of these values can be (and usually are) less than the total number of candidate modeling functions, n_c.

Model parameter uncertainty can be found from Eq. (10),

$$\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\theta}}_m\right) = \hat{\sigma}^2 \left(\boldsymbol{R}_m^T \boldsymbol{R}_m\right)^{-1} \tag{41}$$

where $\boldsymbol{R}_m$ includes only the first m rows and columns of $\boldsymbol{R}$, and m is the index associated with the last selected model term using vector $\boldsymbol{Q}^T\boldsymbol{z}$. The fit error variance estimate is obtained from a running sum of the squared individual residuals left on the right side of Eq. (25) after each recursive orthonormalization,

$$\hat{\sigma}_i^2 = \left[\hat{\sigma}_{i-1}^2 (i-1) + \varepsilon_i^2 \right] / i \tag{42}$$

and $\hat{\sigma}^2$ is initialized from the initial $\boldsymbol{QR}$ decomposition applied to the first n_c data points. Note that the squared residual estimate from Eq. (42) assumes that all n_c candidate modeling functions are included in the model. When $n < n_c$ modeling functions are selected for the model (the usual case), then this squared residual estimate must be augmented with the squared residual associated with the modeling functions that were not selected for the model. Fortunately, this is easy to compute, because the fit error variance reduction associated with each omitted $\boldsymbol{q}_j$ function is simply $\left(\boldsymbol{q}_j^T\boldsymbol{z}\right)^2$, from Eq. (35). So, the fit error variance from Eq. (42) is simply augmented by adding $\left(\boldsymbol{q}_j^T\boldsymbol{z}\right)^2$ for each $\boldsymbol{q}_j$ not selected for the model. Finally, corrections for colored residuals can be made post-flight[12] or using real-time estimates of the residual autocorrelation function[20].

The recursive orthonormalization and automated model structure determination described here identifies which model terms from the candidate pool are necessary to characterize the functional dependencies, and estimates the associated model parameters and uncertainties. Because the approach is based on time-domain equation-error modeling, if data points are missed because of instrumentation or telemetry malfunction, or calculation delays, the modeling algorithm is unaffected (except for the small loss of information in the missed data points), because the equation-error approach in the time-domain does not depend on time sequencing of the data. Therefore, gaps in the real-time data stream can be tolerated without modification to the algorithm. However, large data values associated with dropouts will adversely affect the modeling, as is the case for all time-domain modeling methods.

There are no requirements regarding the form of the candidate modeling functions - they can be multivariate polynomials, multivariate spline functions, or any other linear or nonlinear function that can be computed from the explanatory variable data. Inputs required from the analyst relate only to the limits of what should be considered, such as which explanatory variables to consider, maximum order of multivariate polynomial functions to consider, spline knot locations, and so on. Obviously, the identified model is dependent on the candidate modeling terms available for selection. However, the pool of candidate modeling terms can be specified generously, subject to computational constraints, because the modeling algorithm automatically sorts out which terms are important, based on the data, and omits the rest. The result is a global parsimonious model that characterizes the functional dependencies accurately and predicts well.

V. Flight Test Results

Flight testing was conducted at NASA Wallops Flight Test Facility in Wallops Island, VA, using the instrumented Bat-4 aircraft, shown in Fig. 2 and described in Section II. The flight tests were conducted using the NASA Langley AirSTAR flight testing capability. References [21]-[24] contain information on AirSTAR and associated flight test operations. An overview of the Bat-4 flight testing can be found in Ref. [24].

Real-time global modeling results from flight data for nondimensional aerodynamic force coefficient C_Z are shown in Fig. 5. This figure is a screen shot of the real-time display from a laptop computer inside the AirSTAR Mobile Operations Station (MOS) during flight testing. The top plot on the left shows the model fit to the nondimensional coefficient data as a function of time, and the lower plot on the left shows a three-dimensional plot of the model surface fit to C_Z data (indicated by + markers) as a function of selected explanatory variables, which are angle of attack and elevator deflection. The plots on the right side show time series data for selected explanatory variables. Figure 6 shows a similar display for nondimensional aerodynamic pitching moment coefficient C_m over the same time period. Figures 5 and 6 show the real-time modeling results at the end of the first two of the four sequential Learn-To-Fly maneuvers shown in Fig. 3. The real-time display continuously showed the time evolution of the data and the real-time global nonlinear aerodynamic modeling results.

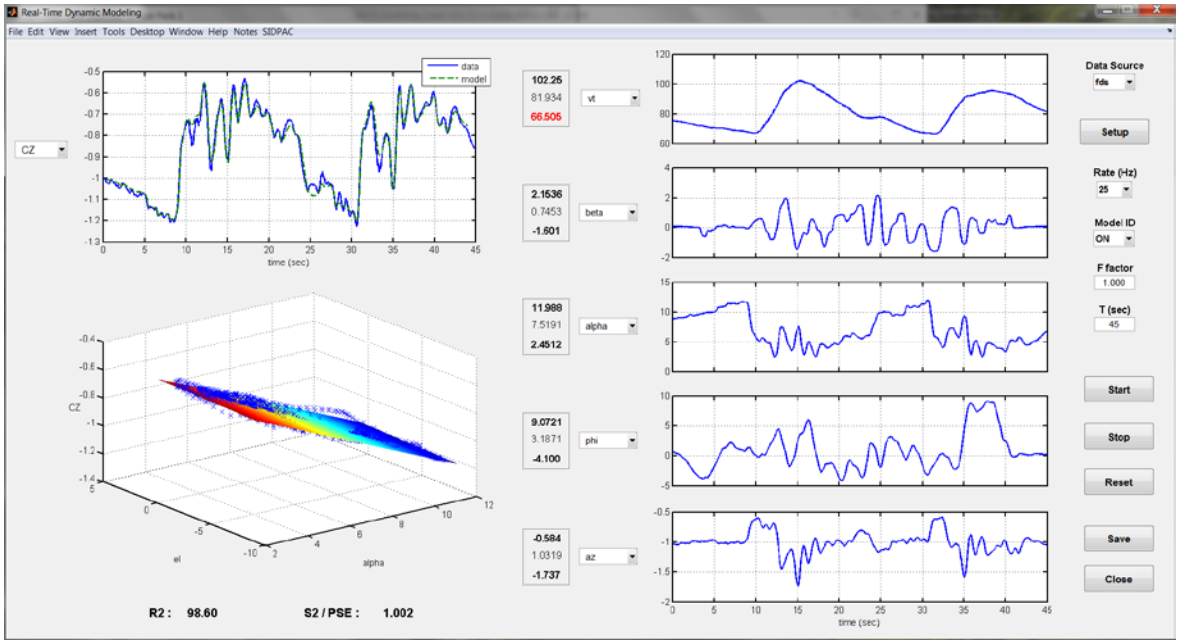

Figure 5. **Data monitoring and real-time global modeling for the CZ coefficient, maneuvers 39a and 39b**

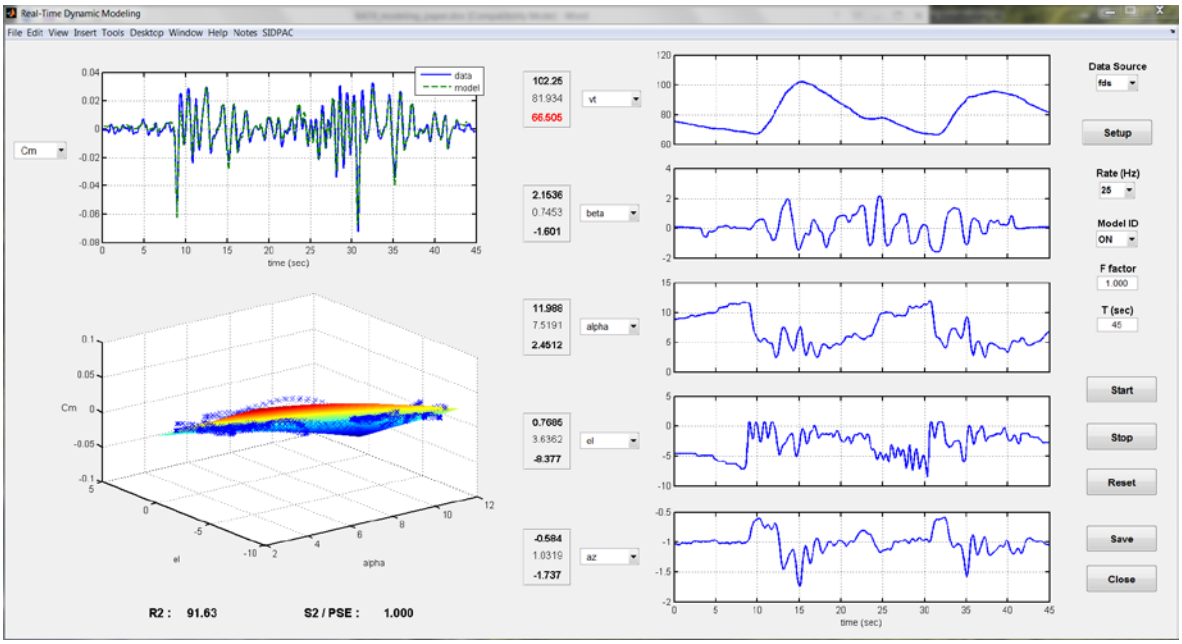

Figure 6. **Data monitoring and real-time global modeling for the Cm coefficient, maneuver 39a and 39b**

Models for the longitudinal nondimensional aerodynamic coefficients were selected from a common pool of candidate longitudinal modeling functions:

$$1,\ \alpha,\ \delta_e,\ \frac{q\bar{c}}{2V},\ \alpha^2,\ \alpha\delta_e,\ \alpha\frac{q\bar{c}}{2V},\ \delta_e\frac{q\bar{c}}{2V},\ \frac{q\bar{c}}{2V}\left|\frac{q\bar{c}}{2V}\right|,\ \delta_e|\delta_e| \tag{43}$$

These candidate functions represent all multivariate polynomials up to 2nd order in the longitudinal explanatory variables α, δ_e, and $\frac{q\bar{c}}{2V}$. The second-order terms in nondimensional pitch rate and elevator deflection used absolute value for one factor, to enforce an odd function for these 2nd order terms. The pool of candidate modeling functions for the lateral nondimensional aerodynamic coefficients was:

$$1,\ \beta,\ \delta_a,\ \delta_r,\ \frac{pb}{2V},\ \frac{rb}{2V},\ \frac{pb}{2V}\frac{rb}{2V},\ \beta\delta_a,\ \beta\frac{pb}{2V},\ \beta\frac{rb}{2V},\ \frac{rb}{2V}\delta_r,\ \frac{rb}{2V}\delta_a,\ \frac{pb}{2V}\delta_a,\ \beta\delta_r,\ \frac{pb}{2V}\delta_r,\ \beta|\beta| \tag{44}$$

This pool of candidates included all linear terms in the lateral explanatory variables β, δ_a, δ_r, $pb/2V$, and $rb/2V$, along with most of the 2nd order terms. Some 2nd order terms that were considered unlikely to be selected for the lateral models were omitted from the candidate pool, to save computation time.

Global models were identified for all six nondimensional aerodynamic coefficients simultaneously in real time. The pool of candidate modeling functions could be made larger and/or specific to each individual nondimensional force or moment coefficient, at the cost of more computation time and computer memory for the recursive orthonormalization and model structure determination.

Recursive orthonormalization of all candidate model terms was done at every time step (25 Hz), but the model structure determination and parameter estimation was done at a reduced 2 Hz rate, to save computation time and because models typically change at a relatively slow rate, provided the aircraft has not incurred damage. All of the data import, filtering, nonlinear modeling calculations, and plotting were done in real time on a Windows® laptop with a 2.7 GHz Intel i7-4800MQ processor running MATLAB® 2009b SP1, using an average 60 percent of the 0.04 s (25 Hz) frame time. Approximately 95 percent of the total computation time was occupied with producing the real-time graphics. More efficient real-time graphics processing would have enabled the use of larger pools of candidate modeling terms; however, the candidate modeling terms specified in (43) and (44) proved adequate for modeling the aerodynamics of the Bat-4 aircraft.

Global models identified for all six nondimensional force and moment coefficients using flight data from the first two Learn-To-Fly maneuvers in Fig. 3 (maneuvers 39a and 39b, approximately 45 s of flight data) were used to predict the flight data for the last two Learn-To-Fly maneuvers shown in Fig. 3 (maneuvers 39c and 39d). The plots in Fig. 7 show the prediction results for all coefficients. The predictions shown in Fig. 7 demonstrate good global model prediction capability for data that were not used to identify the model, with some degradation apparent in the predictions for C_Y and C_l. The same identified models were also applied to predict data from local maneuver 29b, with results shown in Fig. 8. In this case, the C_X prediction is degraded because of uncommanded engine speed changes during the maneuver, which impacted the C_X data and were not included in the C_X model. Otherwise, the identified global model also exhibits good prediction capability for a local maneuver at a specific flight condition. Similar prediction results were obtained for the other local maneuvers 29a and 29c. The prediction results give confidence that the global model structure and associated model parameters were identified accurately. Tables 3 and 4 provide the identified global model terms, model parameter estimates, and associated uncertainties. All angular quantities used in the modeling were expressed in radians. Note that none of the identified model structures included all of the available candidate model terms. The modeling algorithm automatically selected the important model terms, based on the real-time flight data.

Similar analyses were done using real-time modeling results from various combinations of single and multiple Learn-To-Fly maneuvers to predict the data from the remaining Learn-To-Fly maneuvers. Those cases produced similar results to those shown in Figs. 7 and 8, except that the predictions were degraded slightly when the global models were identified from only a single Learn-To-Fly maneuver. However, it was also found that using more than two Learn-To-Fly maneuvers made little improvement in the global model predictions, apart from a small decrease in the estimated parameter uncertainties. This might be because the data from the Learn-To-Fly maneuvers covered similar parts of the explanatory variable subspace, and once a certain level of data coverage was reached, little additional information was added by more Learn-To-Fly maneuvers.

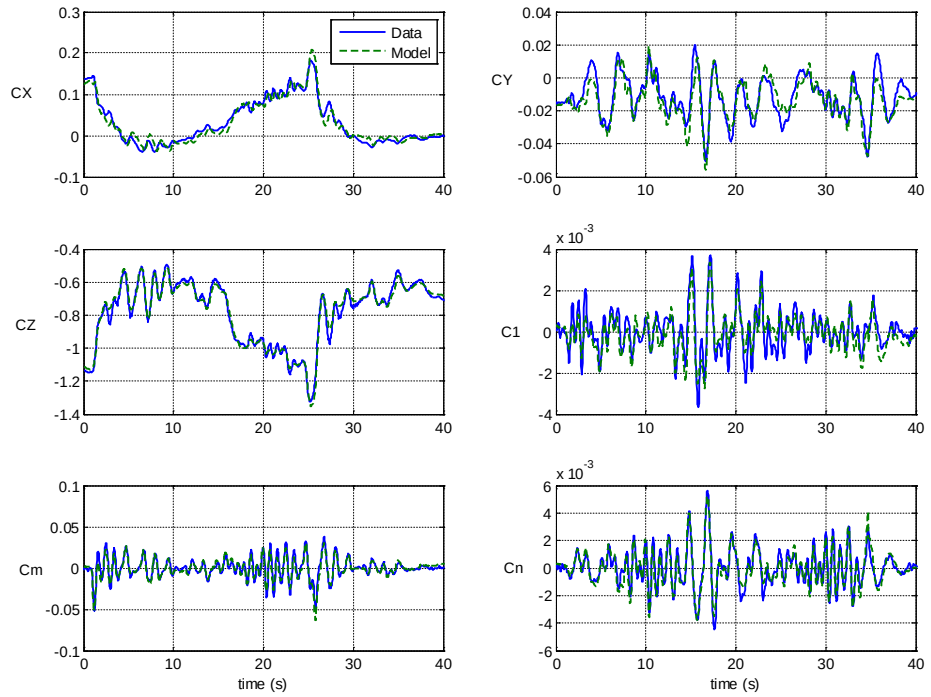

Figure 7. **Aerodynamic predictions for global maneuvers 39c and 39d, using real-time global models identified from maneuvers 39a and 39b.**

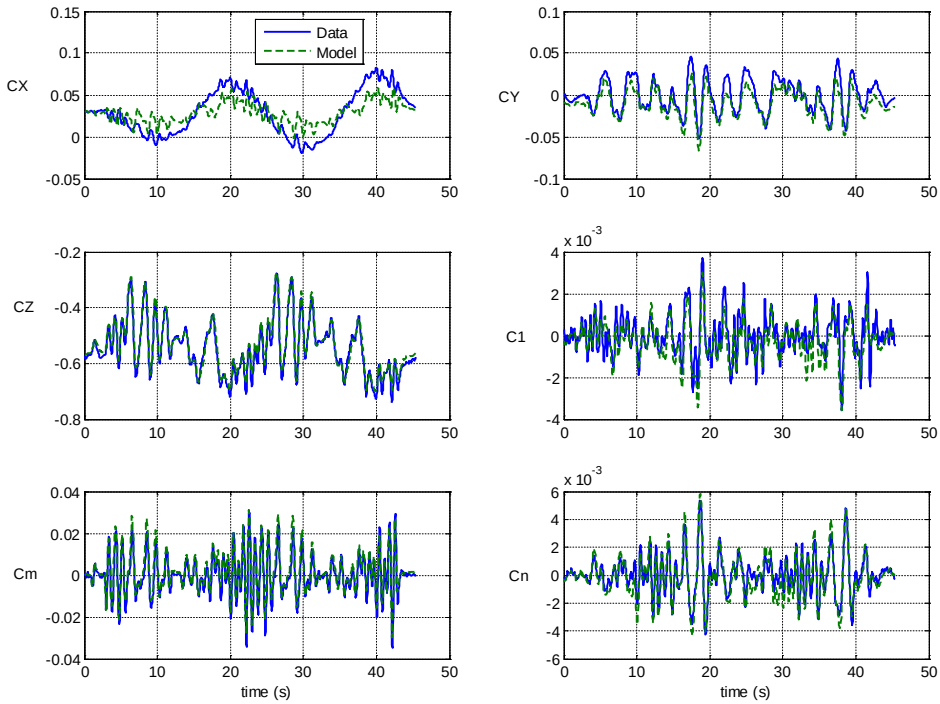

Figure 8. **Aerodynamic predictions for local maneuver 29a, using real-time global models identified from maneuvers 39a and 39b.**

Table 3. Model structure, parameter estimates, and uncertainties for global longitudinal aerodynamic models identified from Bat-4 flight data, maneuvers 39a and 39b

C_X Model Parameter	Estimate ± Standard Error	C_Z Model Parameter	Estimate ± Standard Error	C_m Model Parameter	Estimate ± Standard Error
C_{X_o}	−0.041 ±0.003	C_{Z_o}	−0.390 ±0.002	C_{m_o}	0.004 ±0.001
C_{X_α}	0.263 ±0.052	C_{Z_α}	−3.858 ±0.024	C_{m_α}	−0.151 ±0.016
$C_{X_{\delta_e}}$	−0.175 ±0.018	$C_{Z_{\delta_e}}$	0.141 ±0.035	$C_{m_{\delta_e}}$	−0.599 ±0.006
C_{X_q}	−16.26 ±0.51	C_{Z_q}	−14.48 ±0.98	C_{m_q}	−9.926 ±0.159
$C_{X_{\alpha^2}}$	2.884 ±0.206			$C_{m_{\alpha^2}}$	−1.015 ±0.064

Table 4. Model structure, parameter estimates, and uncertainties for global lateral aerodynamic models identified from Bat-4 flight data, maneuvers 39a and 39b

C_Y Model Parameter	Estimate ± Standard Error	C_l Model Parameter	Estimate ± Standard Error	C_n Model Parameter	Estimate ± Standard Error
C_{Y_o}	−0.0160 ±0.0003	C_{l_o}	0.0007 ±0.0000	C_{n_o}	0.0014 ±0.0000
C_{Y_β}	−0.5008 ±0.0214	C_{l_β}	−0.0306 ±0.0023	C_{n_β}	0.0264 ±0.0017
$C_{Y_{\delta_a}}$	0.2241 ±0.0182	$C_{l_{\delta_a}}$	−0.0615 ±0.0020	$C_{n_{\delta_a}}$	−0.0229 ±0.0015
$C_{Y_{\delta_r}}$	0.0943 ±0.0109	$C_{l_{\delta_r}}$	−0.0097 ±0.0012	$C_{n_{\delta_r}}$	−0.0609 ±0.0009
C_{Y_p}	0.4003 ±0.0763	C_{l_p}	−0.2224 ±0.0082	C_{n_p}	−0.1634 ±0.0063
C_{Y_r}	1.2707 ±0.0542	C_{l_r}	0.2148 ±0.0058	C_{n_r}	−0.0047 ±0.0045
$C_{Y_{\beta\delta_a}}$	−10.296 ±0.987	$C_{l_{\beta\delta_a}}$	−0.8548 ±0.1062		

VI. Conclusions

Research was conducted to develop and demonstrate efficient, autonomous, real-time global nonlinear aerodynamic modeling for aircraft based on flight data alone. Flight data were obtained using flight test maneuvers that simulated dropping the aircraft from altitude, as would be done when implementing a Learn-To-Fly concept. Automated orthogonal optimized multi-axis perturbation inputs were applied throughout the maneuvers at varying flight conditions, resulting in data with rich information content for global nonlinear aerodynamic modeling. Flight data generated with this type of maneuver exhibited low correlations among explanatory variables, multi-axis excitation, and rapid coverage of a large portion of the explanatory variable subspace for aerodynamic modeling, corresponding to a large portion of the aircraft flight envelope. These characteristics make the maneuvers very effective and efficient for global aerodynamic modeling.

A real-time nonlinear global modeling technique was developed by combining multivariate orthogonal function modeling using statistical modeling metrics with real-time recursive modeling function orthonormalization based on the ***QR*** decomposition and Givens rotations. The technique identified global aerodynamic models for all six rigid-body degrees of freedom simultaneously in real time from flight data alone, and achieved excellent model fits to the data. The resulting global aerodynamic models exhibited good prediction capability for flight data from both global and local flight maneuvers that were not used in the modeling process.

Investigations will continue to evaluate data information content from global flight test maneuvers and the quality of the identified global aerodynamic models as a function of time and data information. These issues are important for successful integration with learning adaptive control and guidance, which will be critical for the success of the Learn-To-Fly concept. Learn-To-Fly technology will enable self-learning aircraft, resulting in reduced development cost for new aircraft, and more robust and safe flight operations. The flight testing and modeling techniques described in this work can also be applied to build high-fidelity, full-envelope nonlinear aircraft simulations from flight data, thus reducing or eliminating the need for extensive wind tunnel testing or aerodynamic calculations.

Acknowledgments

The efforts of the AirSTAR flight test team at NASA Langley Research Center in modifying and testing the Bat-4 aircraft and associated systems, carefully calibrating the instrumentation, and carrying out the flight operations to collect the high-quality flight data used in this study, are gratefully acknowledged. The flight operations and safety personnel at NASA Wallops were essential for conducting the flight testing in an efficient and safe manner. Research in Aircraft System Identification is funded by the NASA Aeronautics Research Mission Directorate (ARMD) Learn-To-Fly project.

References

[1]McNally, B. David, "Full-Envelope Aerodynamic Modeling of the Harrier Aircraft," NASA TM 88376, October 1986.

[2]Jategaonkar, R.V., Mönnich, W., Fischenberg, D. and Krag, B. "Identification of C-160 Simulator Data Base from Flight Data," *Proceedings of the 10th IFAC Symposium on System Identification*, Copenhagen, Denmark, Elsevier Sciences Ltd., Oxford, UK, 1994, pp. 1031-1038.

[3]Hui, K., Ricciardi, J., Ellis, K, and Tuomey, D. "Beechjet Flight Test Data Gathering and Level-D Simulator Aerodynamic Mathematical Model Development," AIAA 2001-4012, *AIAA Atmospheric Flight Mechanics Conference*, Montreal, Quebec, Canada, August 2001.

[4]Seher-Weiss, S. "Identification of Nonlinear Aerodynamic Derivatives using Classical and Extended Local Model Networks," *Aerospace Science and Technology*, Vol. 15, 2011, pp. 33-44.

[5]Tobias, E.L., Tischler, M.B., Berger, T., and Hagerott, S.G. "Full Flight-Envelope Simulation and Piloted Fidelity Assessment of a Business Jet Using a Model Stitching Architecture," *AIAA Modeling and Simulation Technologies Conference*, Kissimmee, FL, January 2015.

[6]Morelli, E.A. "Flight Test Maneuver Design for Efficient Aerodynamic Modeling," AIAA-2011-6672, *AIAA Atmospheric Flight Mechanics Conference*, Portland, OR, August 2011.

[7]Morelli, E.A. "Efficient Global Aerodynamic Modeling from Flight Data," AIAA-2012-1050, *50th AIAA Aerospace Sciences Meeting*, Nashville, TN, January 2012.

[8]Brandon, J.M. and Morelli, E.A. "Nonlinear Aerodynamic Modeling From Flight Data Using Advanced Piloted Maneuvers and Fuzzy Logic," NASA/TM-2012-217778, October 2012.

[9]Morelli, E.A., Cunningham, K., and Hill, M.A. "Global Aerodynamic Modeling for Stall/Upset Recovery Training Using Efficient Piloted Flight Test Techniques," AIAA 2013-4976, *AIAA Modeling and Simulation Technologies Conference*, Boston, MA, August 2013.

[10]Brandon, J.M. and Morelli, E.A. "Real-Time Global Nonlinear Aerodynamic Modeling from Flight Data," AIAA-2014-2554, *AIAA Atmospheric Flight Mechanics Conference*, Atlanta, GA, June 2014.

[11]Morelli, E.A. "Global Nonlinear Aerodynamic Modeling using Multivariate Orthogonal Functions," *Journal of Aircraft*, Vol. 32, No. 2, March-April 1995, pp. 270-77.

[12]Klein, V. and Morelli, E.A., *Aircraft System Identification – Theory and Practice*, AIAA Education Series, AIAA, Reston, VA, 2006.

[13]Gentleman, W.M. "Least Squares Computation by Givens Transformation Without Square Roots," *Journal of the Institute for Mathematics Applications*, Vol. 12, 1973, pp. 329-336.

[14]Gentleman, W.M. "Regression Problems and the QR Decomposition," *Bulletin of the Institute for Mathematics Applications*, Vol. 10, 1974, pp. 195-197.

[15]Luo, W., Billings, S.A., and Tsang, K.M. "On-line Structure Detection and Parameter Estimation with Exponential Windowing for Nonlinear Systems," *European Journal of Control*, Vol. 2, 1996, pp. 291-304.

[16]http://www.sunflyte.com/SIDBook.html

[17]Klein, V. and Batterson, J.G. "Determination of Airplane Model Structure From Flight Data Using Splines and Stepwise Regression," NASA TP-2126, March 1983.

[18]Morelli, E.A., "Practical Aspects of the Equation-Error Method for Aircraft Parameter Estimation," AIAA-2006-6144, *AIAA Atmospheric Flight Mechanics Conference*, Keystone, CO, August 2006.

[19]Barron, A.R., "Predicted Squared Error : A Criterion for Automatic Model Selection," *Self-Organizing Methods in Modeling*, Farlow, S.J., Ed., Marcel Dekker, Inc., New York, NY, 1984, pp. 87-104.

[20]Holzel, M.S. and Morelli, E.A. "Real-Time Frequency Response Estimation from Flight Data," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, September-October 2012, pp. 1406-1417.

[21]Jordan, Thomas L., Langford, William M., and Hill, Jeffrey S.; "Airborne Subscale Transport Aircraft Research Testbed: Aircraft Model Development", AIAA 2005-6432, *AIAA Guidance, Navigation, and Control Conference and Exhibit,* San Francisco, CA, August 2005.

[22]Jordan, Thomas L., Foster, John V., Bailey, Roger M., and Belcastro, Christine M., "AirSTAR: A UAV Platform for Flight Dynamics and Control System Testing," AIAA-2006-3307, *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, San Francisco, CA, June 2006.

[23]Murch, A. M., "A Flight Control System Architecture for the NASA AirSTAR Flight Test Facility," AIAA 2008-6990, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Honolulu, HI, 2008.

[24]Cunningham, K., Cox, D., Foster, J., Riddick, S., and Laughter, S., "AirSTAR Beyond Visual Range System Description and Preliminary Test Results," *AIAA SciTech Conference*, San Diego, CA, January 2016.