Real-Time Parameter Estimation Using Output Error

Jared A. Grauer[*]

NASA Langley Research Center, Hampton, Virginia, 23681

Output-error parameter estimation, normally a post-flight batch technique, was applied to real-time dynamic modeling problems. Variations on the traditional algorithm were investigated with the goal of making the method suitable for operation in real time. Implementation recommendations are given that are dependent on the modeling problem of interest. Application to flight test data showed that accurate parameter estimates and uncertainties for the short-period dynamics model were available every 2 s using time domain data, or every 3 s using frequency domain data. The data compatibility problem was also solved in real time, providing corrected sensor measurements every 4 s. If uncertainty corrections for colored residuals are omitted, this rate can be increased to every 0.5 s.

Nomenclature

Roman

$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	linear system matrices
a_x, a_y, a_z	body translational accelerations [ft/s^2]
cov	covariance
g	gravitational acceleration [ft/s^2]
$\mathbb{I}$	identity matrix
j	imaginary number, $\sqrt{-1}$
p, q, r	body roll, pitch, and yaw rates [rad/s]
$\bar{q}$	dynamic pressure [lbf/ft^2]
t	time [s]
u, v, w	body translational velocities [ft/s]
V	airspeed [ft/s]

Greek

α	angle of attack [rad]
β	sideslip angle [rad]
γ	flight path angle [rad]
Δ	perturbation value
δ_e	elevator deflection [rad]
σ	standard deviation
ϕ, θ, ψ	Euler angles [rad]
ω	frequency [rad/s]

Superscripts

$^{-1}$	matrix inverse
^{T}	transpose
$\dot{}$	time derivative
†	complex conjugate transpose
$\hat{}$	estimated value

Subscripts

$_0$	trim or starting value

[*] Research Engineer, Dynamic Systems and Control Branch, MS 308, Member AIAA

I. Introduction

Output error has matured into one of the most widely accepted and used methods in the aerospace industry for estimating unknown model parameters from measured flight data. It is a straightforward method based on maximum likelihood estimation that uses realistic assumptions consistent with standard flight test practices and the real physical situation, and also possesses useful analytic properties. These favorable characteristics have resulted in successful application to a broad range of research problems including aerodynamic modeling, control law verification and validation, envelope expansion, flight path reconstruction, and accident investigation. Popularization of this method is due, in part, to numerous researchers releasing software packages that combine the rigorous estimation theory with pragmatic numerical methods and practical engineering experience.[1,2,3,4,5]

Aircraft system identification is currently undergoing a paradigm shift towards real-time modeling during flight. This effort is facilitated by improvements in hardware and computing capability, and driven by the need to reduce costs and improve safety. For example, immediate feedback of modeling quality during control law flight tests can lead to shorter maneuver lengths, or can diagnose problems while on condition, thereby saving time and costs while avoiding subsequent tests. Safety issues, such as icing or fault-detection, can be addressed in real time by monitoring estimation results.[6]

There are however several aspects of output error that make a real-time implementation difficult. Output error is a parametric modeling technique that requires prior knowledge of a model structure. This requirement is problematic if there is little or no information before a flight test or if the dynamics change, for example because of structural damage to the aircraft. Another difficulty is that output error is a nonlinear estimation problem that uses an iterative multi-dimensional nonlinear optimization method to determine the solution. These iterative methods not only require more computation time than non-iterative solutions, but they require good starting values for the unknown model parameters. Poor starting values can lead to excessive iterations, incorrect solutions, or divergence. Another problem is that after a solution is found, additional effort is needed to correct uncertainty estimates for colored residuals, which appear in practice because of unmodeled dynamics but are not assumed in the theory.[5,7,8] Finally, it is exceedingly difficult to apply traditional output error to unstable aircraft, such as fighters or damaged aircraft, because small errors in the model parameters and initial conditions quickly lead to divergent time histories of the model outputs.[5,9]

The closest attempt to a real-time implementation of output error was by Hui et al.,[10] who used FORTRAN software onboard a Dassault Falcon 20 to produce parameter estimates in "near real time," soon after a maneuver completed using a batch process. Other researchers have looked to alternative methods for real-time parameter estimation. Filter-error and the extended Kalman filter have been used, especially when turbulence is present,[11,12] but involve tuning parameters and produce inaccurate uncertainties. Equation-error parameter estimation in the frequency domain has been applied successfully in real time,[13] but this method requires an extra step in the time domain to estimate bias parameters, and does not directly optimize the model parameters to fit measured output data in the time domain. A real-time frequency response method can be used without knowing the model structure,[14] but additional effort is needed to deduce the structure from the results and estimate parameters. Currently there is no working real-time parameter estimation technique that uses the output-error approach.

This paper addresses some of the shortcomings of output error, to make it suitable for solving real-time parameter estimation problems. Variations to the method are implemented and tested to determine best practices, and then results are applied to flight test data to estimate model parameters in real time. Section II presents traditional output error and its variations. A subscale flight test vehicle and its nonlinear flight simulation are described in Section III, as well as a maneuver used for system identification. Output-error results are presented in Section IV for the real-time estimation of short period model parameters and for solving the data compatibility problem during flight. Discussion is provided in Section V.

II. Methods

A. Traditional Output Error

This section introduces traditional output error. For a comprehensive description, see Refs. [5, 15]. Consider the continuous-time model of the aircraft flight dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}\right), \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{1}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the applied control input, and $\boldsymbol{\theta}$ is a vector of n_p unknown model parameters. Disturbances in the state equations, e.g. due to turbulence or model structure errors, are neglected, and the control input is assumed to be measured without error. Consider also the n_0 discrete-time sensor measurements

$$\mathbf{z}(i) = \mathbf{y}(i) + \boldsymbol{\nu}(i), \qquad i = 1, 2, \ldots, N \tag{2}$$

observed at N regular time intervals, where $\mathbf{y}$ is the model output and $\boldsymbol{\nu}$ is a Gaussian white noise sequence with zero mean and covariance $\mathbf{R}$.

The objective of output error is to determine the unknown model parameters that best match the model output to the measured output. This is done by minimizing the cost function

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{v}(i)^T \mathbf{R}^{-1} \boldsymbol{v}(i) \tag{3}$$

where

$$\boldsymbol{v}(i) = \mathbf{z}(i) - \mathbf{y}(i) \tag{4}$$

is the modeling residual. Optimization requires an iterative method because the relationship between the model parameters and the model outputs is nonlinear. This process is outlined as follows:

1. Solve the equations of motion using the measured input, initial state, and the initial parameter estimate $\hat{\boldsymbol{\theta}}_0$ to compute the model states and outputs.

2. Compute the model residual and estimate the noise covariance matrix

$$\hat{\mathbf{R}} = \operatorname{diag}\left(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}(i)\boldsymbol{v}(i)^T\right) \tag{5}$$

3. Compute the output-error cost function, as well as the output sensitivity, cost gradient, and information matrices

$$\mathbf{S}(i) = \left\{\frac{\partial y_a(i)}{\partial \theta_b}\right\}, \qquad \text{for } a = 1, 2, \ldots, n_o \text{ and } b = 1, 2, \ldots, n_p \tag{6}$$

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = -\sum_{i=1}^{N} \mathbf{S}(i)^T \hat{\mathbf{R}}^{-1} \boldsymbol{v}(i) \tag{7}$$

$$\mathbf{M} = \sum_{i=1}^{N} \mathbf{S}(i)^T \hat{\mathbf{R}}^{-1} \mathbf{S}(i) \tag{8}$$

4. Update the parameter estimates using the modified Newton-Raphson method[1,5] and compute the Cramér-Rao bounds

$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_0 - \mathbf{M}^{-1}\frac{\partial J}{\partial \boldsymbol{\theta}} \tag{9}$$

$$\operatorname{cov}(\hat{\boldsymbol{\theta}}) \geq \mathbf{M}^{-1} \tag{10}$$

5. Repeat until the solution converges

6. Correct parameter uncertainties for colored residuals.[5,7,8]

B. Frequency Domain Estimation

Output error can be reformulated to use frequency domain data.[5,16] Measurements are converted into the frequency domain by removing the bias and trend, to avoid low-frequency leakage, and then applying a high-accuracy chirp-z Fourier transform.[5,17] This step restricts the equations of motion to linear systems

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= \mathbf{A}(\boldsymbol{\theta})\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\theta})\mathbf{u}(t), \qquad \mathbf{x}(t_0) = \mathbf{0} \\
\mathbf{y}(t) &= \mathbf{C}(\boldsymbol{\theta})\mathbf{x}(t) + \mathbf{D}(\boldsymbol{\theta})\mathbf{u}(t)
\end{aligned} \tag{11}$$

where the states, control inputs, and model outputs represent perturbations from their respective trim values. The system output in the frequency domain is

$$\mathbf{y}(j\omega) = \left[\mathbf{C}(j\omega\mathbb{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right]\mathbf{u}(j\omega) \tag{12}$$

and the cost function to be minimized is

$$J(\boldsymbol{\theta}) = N_f \sum_{k=1}^{N_f} \boldsymbol{\upsilon}(j\omega_k)^{\dagger}\mathbf{S}_{\nu\nu}^{-1}\boldsymbol{\upsilon}(j\omega_k) \tag{13}$$

where $\mathbf{S}_{\nu\nu}$ is the power spectral density of the noise and N_f is the number of frequencies. The output-error algorithm then proceeds as normal.

The most restrictive drawback of using output error in the frequency domain is that only linear systems may be considered. Because linear equations describe perturbation dynamics and because of the need to detrend the time-domain data prior to Fourier transformation (to avoid leakage), bias parameters cannot be estimated in the frequency domain. In addition, practical experience using output error with frequency domain data has exhibited less robustness to model parameter starting values and has required more iterations for convergence.

Offsetting these disadvantages, there are several advantages to using frequency domain data. The Fourier transform can be applied at arbitrary frequencies, so that only the data information relevant to the dynamics of interest is transformed. This typically results in many fewer data points than in the original time series, yielding faster computations. Because only the frequency content of interest is transformed, colored residual corrections to the uncertainties are unnecessary. In the frequency domain, integration is replaced by multiplication, which is computationally faster and makes identifying an unstable dynamic system no more difficult than identifying a stable dynamic system. These benefits make the use of frequency domain data viable for real-time implementation of output error.

C. Output Sensitivities

The output sensitivities reflect how the model parameters change the model outputs, and strongly affect the convergence rate and solution. A general method to compute the sensitivities is to use finite differences. This method can be applied to arbitrary systems with no change in software. The detriment of doing this is that a large number of time-consuming integrations are performed when using time domain data.

An alternative method is to evaluate analytically-derived sensitivity equations. Although these equations are difficult to derive for a general system, for a linear system they are

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}}\right) &= \mathbf{A}\frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}}\mathbf{x} + \frac{\partial \mathbf{B}}{\partial \boldsymbol{\theta}}\mathbf{u}, \qquad \frac{\partial \mathbf{x}(t_0)}{\partial \boldsymbol{\theta}} = \mathbf{0} \\
\frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} &= \mathbf{C}\frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}}\mathbf{x} + \frac{\partial \mathbf{D}}{\partial \boldsymbol{\theta}}\mathbf{u}
\end{aligned} \tag{14}$$

The corresponding representation in the frequency domain is

$$\frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} = \mathbf{C}\left(j\omega\mathbb{I} - \mathbf{A}\right)^{-1}\left(\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}}\mathbf{x}(j\omega) + \frac{\partial \mathbf{B}}{\partial \boldsymbol{\theta}}\mathbf{u}(j\omega)\right) + \left(\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}}\mathbf{x}(j\omega) + \frac{\partial \mathbf{D}}{\partial \boldsymbol{\theta}}\mathbf{u}(j\omega)\right) \tag{15}$$

This approach involves additional effort to derive and program the sensitivity equations. This effort could be considerable when applying to nonlinear systems or when several candidate model structures are being evaluated. However, the sensitivities are computed approximately twice as fast using the supplied equations rather than numerical finite differences. Additionally, when using noisy data, supplying the model structure of the sensitivity equations has a smoothing effect on the sensitivities and therefore helps the nonlinear optimization used in output error parameter estimation. These characteristics are well-suited for real-time implementation.

D. Parameter Initialization

Good starting values of the model parameters are required for timely convergence; otherwise, the algorithm could use an excessive number of iterations, converge on an incorrect solution, or diverge. Sources for starting values may include previous experiments, wind tunnel tests, an equation-error analysis, or experience.

Denery presented a method in which the traditional output-error algorithm can be initialized at arbitrary values and still obtain quick convergence.[18] State histories obtained from measurements, instead of integration of the equations of motion, are substituted into the analytically-derived sensitivity equations. The parameter estimates then quickly converge to a neighborhood near the solution.

In addition to adding code complexity, this method requires the sensitivity equations to be analytically derived and the state vector to either be measured directly or reconstructed from measurements. Any noise or error on the substituted measurements will bias the parameter estimates. However, using this method for the first iteration and then switching back to integrating the equations of motion for the remaining iterations removes the requirement for good starting values. Practical experience showed that the method did not perform as well when using frequency domain data. This is expected to be due to the fact that only Fourier transforms are matched, which is the same reason unstable aircraft may be identified using frequency domain data with no additional effort.

E. Sequential Estimation

Output error can provide real-time estimates by operating sequentially on recent measurements. Time-domain data is divided into segments of duration τ. While the data system is collecting the next segment of data, output error is used to analyze the current segment of data. Parameter estimates and uncertainties from the analysis on the previous segment of data is used as prior knowledge for the parameters in a Bayesian approach. The output-error cost function, cost gradient, information matrix, and Newton-Raphson updates to the parameter estimates are formulated as[5]

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{v}(i)^T \mathbf{R}^{-1} \boldsymbol{v}(i) + \frac{1}{2}\left(\boldsymbol{\theta} - \boldsymbol{\theta}_p\right)^T \boldsymbol{\Sigma}_p^{-1}\left(\boldsymbol{\theta} - \boldsymbol{\theta}_p\right) \tag{16}$$

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = -\sum_{i=1}^{N} \mathbf{S}(i)^T \hat{\mathbf{R}}^{-1} \boldsymbol{v}(i) - \boldsymbol{\Sigma}_p^{-1}\left(\boldsymbol{\theta} - \boldsymbol{\theta}_p\right) \tag{17}$$

$$\mathbf{M} = \sum_{i=1}^{N} \mathbf{S}(i)^T \hat{\mathbf{R}}^{-1} \mathbf{S}(i) + \boldsymbol{\Sigma}_p^{-1} \tag{18}$$

$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_0 - \left(\mathbf{M} + \boldsymbol{\Sigma}_p^{-1}\right)^{-1}\left(\frac{\partial J}{\partial \boldsymbol{\theta}} + \boldsymbol{\Sigma}_p^{-1}\left(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_p\right)\right) \tag{19}$$

where $\boldsymbol{\theta}_p$ and $\boldsymbol{\Sigma}_p$ are the parameter estimates and covariance matrices from the previous analysis. In this manner, information from the past is retained. Data forgetting techniques could also be applied here to discard past data and more quickly adapt to changing dynamics.

In the frequency domain, the output data can be transformed using a recursive Fourier transform[5,13] for each frequency ω,

$$\mathbf{z}(j\omega)_i = \mathbf{z}(j\omega)_{i-1} + \mathbf{z}(i)e^{-j\omega i\Delta t}\Delta t \tag{20}$$

and similarly for the input data, where Δt is the sampling period. An output-error analysis is applied periodically using the Fourier transform data in the frequency domain. Parameter estimates from the previous analysis are used as starting values, but the previous uncertainty is not used because Fourier transforms are cumulative and this would more heavily weight past information.

Applying output error sequentially yields parameter and uncertainty estimates while flight tests are being conducted. There is a trade-off in choosing the size of the data segments. For large segments, sequential estimation approaches the traditional post-flight method. For short segments, the information content in each data segment is reduced, as well as the amount of time afforded for the output-error analysis, which could lead to bad starting parameters and initial conditions, or divergence of the estimator. A good compromise is a length that both has enough information content and that provides updates frequently. In this paper, the time length τ is determined using trial and error with simulation and flight test data.

III. Materials

A. Flight Test Aircraft and Nonlinear Flight Simulation

The test aircraft used is known as the T-2, which is shown in Figure 1 and described in Table 1. It is a 5.5% dynamically-scaled version of a generic transport airplane that is used in the NASA Aviation Safety Program flight test capability called AirSTAR. Typical flight operations begin with a safety pilot taking the aircraft to altitude. Control is then transferred to a research pilot, who flies it from a mobile control room using synthetic vision drawn from telemetry data and a database of the local terrain. Control deflection commands are generated from the research pilot and the ground-based flight control system. Computer-generated excitation inputs for system identification are added to these commands just before the position and rate limiters at the actuators.

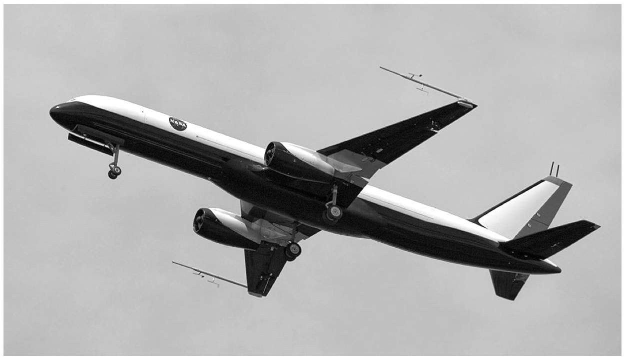


Figure 1. T-2 subscale model (credit: NASA Langley Research Center)

Table 1. GTM mass and geometry values

Parameter	Symbol	value	Units
longitudinal center of mass position	x_{cm}	0.350	$\%\bar{c}$
mean aerodynamic chord	$\bar{c}$	0.915	ft
wing span	b	6.849	ft
wing reference area	S	5.902	ft^2
mass	m	1.542	slug
roll inertia	I_{xx}	1.327	$\text{slug}\cdot\text{ft}^2$
pitch inertia	I_{yy}	4.254	$\text{slug}\cdot\text{ft}^2$
yaw inertia	I_{zz}	5.454	$\text{slug}\cdot\text{ft}^2$
roll/yaw coupling inertia	I_{xz}	0.120	$\text{slug}\cdot\text{ft}^2$

The aircraft has 16 independent control surfaces. For the present study, the controls were deflected in the conventional manner as elevator, aileron, and rudder. Measurements include body-axis translational accelerations and angular rates, estimated Euler angles, differential GPS position and velocity, air flow angles from vanes mounted on wingtip booms, static and dynamic pressure, fuel flow rate, and control surface deflections. First-order anti-aliasing filters with cut-off frequencies at 16 Hz were used before measurement digitization. Data were telemetered to the ground station at 200 Hz, and downsampled to 50 Hz for real-time modeling. For additional information about the aircraft and flight operations, see Refs. [19, 20].

A wind tunnel test article with similar mold lines was studied extensively in the NASA Langley 14- by 22-Foot Subsonic tunnel.[21] Static and forced-oscillation data were used to build a high-fidelity nonlinear flight simulation called the Generic Transport Model (GTM). A modified version of this simulation,[5, 22] containing a smoothed aerodynamic database generated using orthogonal functions, was employed to simulate flight data. This simulation was written in MATLAB®and can be run in batch mode with specified inputs or flown in real time with a pilot stick and visualization software.

B. Flight Maneuver

The results in this paper use flight test data from a single maneuver that was flown using the T-2 aircraft. The measured control surface deflections from this data were supplied as input to the GTM simulation to also produce simulated data for that flight. The reference condition is straight and level flight with true airspeed 139 ft/s, angle of attack 4.0 deg, and altitude 1226 ft.

Orthogonal phase-optimized multisines, an input designed and used extensively at NASA Langley Research Center,[5,23] were employed to excite the aircraft for system identification. These inputs implement simultaneous and uncorrelated excitation on each input over a frequency range of interest and in a manner that keeps the aircraft within small perturbations of the trim condition. It is an efficient technique that has led to successful modeling results in a variety of unusual flight conditions including hypersonic, high sideslip, stall and post-stall flight, as well as normal flight conditions.[23,24]

The measured flight test data from the T-2 aircraft, and also the simulation data from the GTM, are shown in Figure 2. Only the longitudinal measurements are shown here, but the lateral and direction axes of the aircraft are also being excited in a similar but uncorrelated manner. The elevator input contains a 10 s excitation with frequencies between 0.2 to 2.2 Hz in 0.3 Hz increments. The input also contains additional contributions from the pilot and flight control system. While the responses occur at similar frequencies, there are large discrepancies between the flight test and simulation data due to disturbances, differences in the test articles, and under-prediction of the damping values in the simulation model. A global Fourier smoothing technique[5] was applied to the flight data to estimate noise covariances, which were then used for the simulation data.

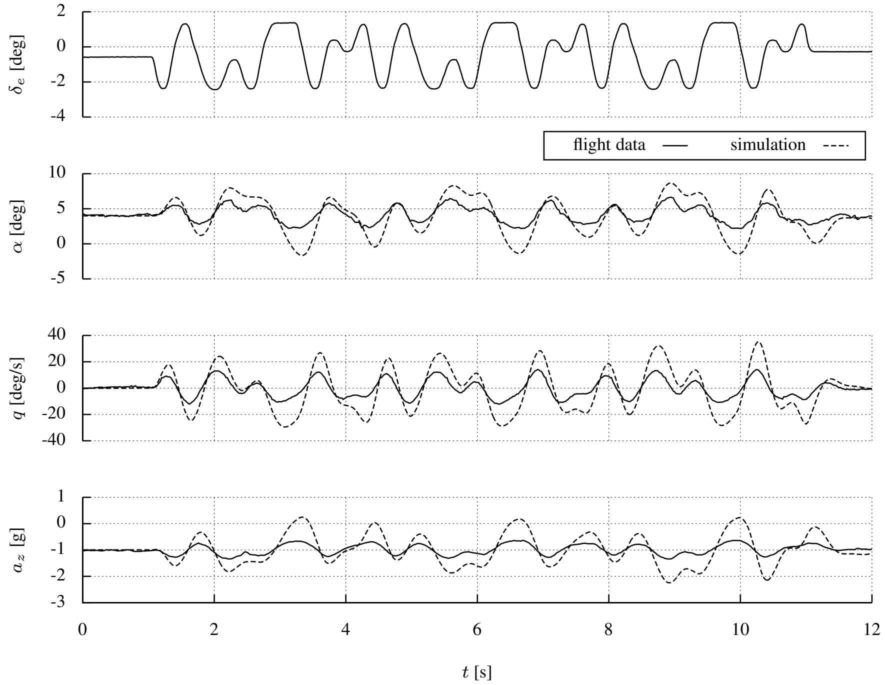


Figure 2. Longitudinal T-2 flight test data and GTM simulation data

C. Software Development

SIDPAC is a collection of computer programs written in MATLAB®[25] at NASA Langley Research Center. This package is bundled with Ref. [5] and contains numerous routines for system identification including experiment design, data conditioning, data compatibility analysis, model structure determination, parameter estimation, and real-time parameter estimation. Specifically, output-error parameter estimation routines using both time and frequency domain data are included in SIDPAC. These codes have evolved from previous NASA software,[1] new algorithms,[5,13,16] and practical experience applying these methods to flight test campaigns at NASA Langley Research Center.

The SIDPAC output-error codes were modified to perform the techniques in Section II not already included in the codes. Output sensitivities can be computed using numerical forward or central finite-differences, or now evaluated using supplied equations. The Denery technique for substituting measured or reconstructed state values can be used, and the code can switch back to integrating the state equations after a specified number of iterations. The code was already capable of sequential estimation, but timer functions were added to limit the execution time during real-time analysis.

Although real-time operation was achieved, there are several options for increasing code execution speed. Results presented herein were performed on a 2.66 GHz laptop. The code could be run on computers with more and faster processors. The code could be further streamlined for computational efficiency, and parallel computing could be employed to make efficient use of multiple processors. Functions could be coded and compiled as MEX-files for increased speed. Alternatively, the entire program could be rewritten in a compiled language such as FORTRAN or C for faster speeds. These options were not explored here because the MATLAB®version of the code is popular, portable, and can be rapidly changed. Another method for quicker convergence times is to change convergence metrics. Currently, there are four metrics that check for convergence of residuals, model parameter estimates, cost, and cost sensitivity to within specific threshold values. These values are currently set conservatively, and may be relaxed to achieve convergence in many fewer iterations but with nearly the same solution.

IV. Results

A. Batch Estimation of Short Period Mode

Output error was first applied to the GTM simulation data in a post-flight analysis to evaluate computation time, accuracy, and overall usefulness of the traditional method and its variations. Model parameters in the short period approximation

$$\begin{aligned}
\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} &= \begin{bmatrix} -\frac{\bar{q}S}{mV_0}C_{L_\alpha} + \frac{g\sin(\gamma_0)}{V_0} - \frac{T_0\cos(\alpha_0)}{mV_0} & 1 - \frac{\bar{q}S\bar{c}}{2mV_0^2}C_{L_q} \\ \frac{\bar{q}S\bar{c}}{I_{yy}}C_{m_\alpha} & \frac{\bar{q}S\bar{c}^2}{2V_0 I_{yy}}C_{m_q} \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} -\frac{\bar{c}S}{mV_0}C_{L_{\delta_e}} & \theta_7 \\ \frac{\bar{q}S\bar{c}}{I_{yy}}C_{m_{\delta_e}} & \theta_8 \end{bmatrix} \begin{bmatrix} \delta_e \\ 1 \end{bmatrix} \\
\begin{bmatrix} \alpha \\ q \\ a_z \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{\bar{q}S}{mg}C_{L_\alpha} & -\frac{\bar{q}S\bar{c}}{2V_0 mg}C_{m_q} \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{\bar{q}S}{mg}C_{L_{\delta_e}} & \theta_9 \end{bmatrix} \begin{bmatrix} \delta_e \\ 1 \end{bmatrix}
\end{aligned} \tag{21}$$

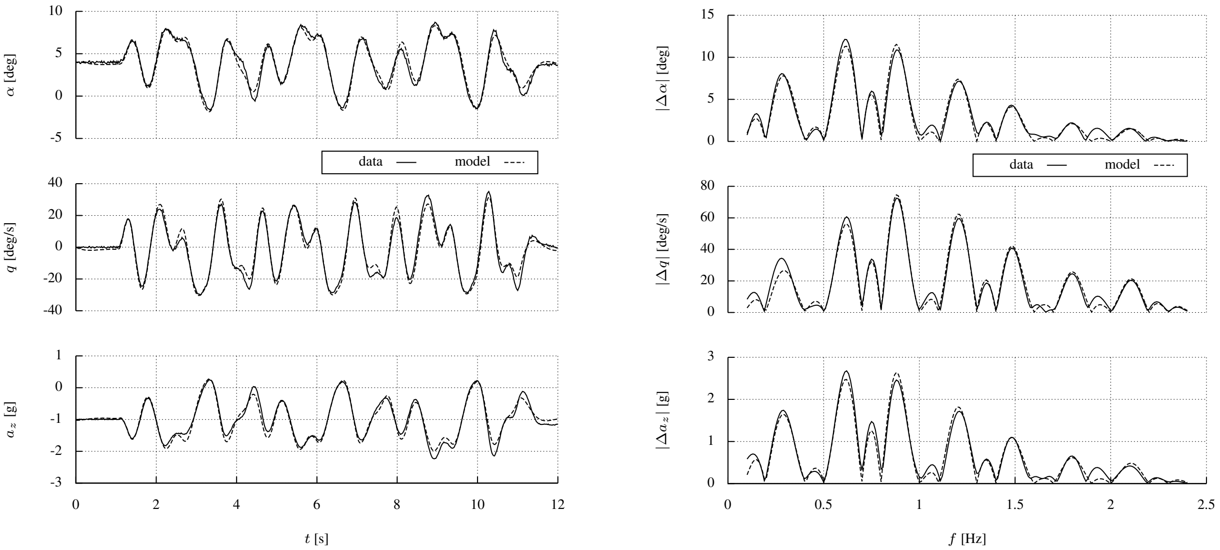
were estimated, where flight variables inside the matrices represent the trim values at the start of the maneuver. The input vector is augmented with a "1" so that bias terms may be estimated. The model parameter vector is

$$\boldsymbol{\theta} = \begin{bmatrix} C_{L_\alpha} & C_{L_q} & C_{L_{\delta_e}} & C_{m_\alpha} & C_{m_q} & C_{m_{\delta_e}} & \theta_7 & \theta_8 & \theta_9 \end{bmatrix}^T \tag{22}$$

where the last three parameters contain aerodynamic bias terms and trim variables. These bias terms are removed from the equation and perturbation states are used when parameter estimation is performed with frequency domain data.

Each analysis was repeated 200 times with unique noise sequences. The initial parameter vector was chosen as $\boldsymbol{\theta}_0 = [1, 1, 1, -1, -1, -1, 0, 0, 0]^T$ to start the stability and control parameters with the correct sign. Typical fits to the measured outputs are shown in the time domain and the frequency domain in Figure 3. In general, the matches were very close, having coefficients of determination above 95% and Theil inequality coefficients below 0.06. The slight mismatches between the model and the data may be due to the excitation of other modes and small nonlinearities in the data. In all the cases good fits were obtained and were in statistical agreement with the numerical linearization solution.

Parameter estimation results are presented in Table 2, which shows estimated parameters and standard errors for C_{L_α} and θ_7, computer execution time, number of iterations until convergence, and time that each iteration used. All estimation results were in statistical agreement with the finite difference solutions. Again, bias parameters could not be estimated using frequency domain data. Cases 1–4 were performed in the time domain, denoted "time," whereas cases 5–6 were in the frequency domain, denoted "freq." The output sensitivities were computed using finite differences in cases 1 and 5, denoted "numerical," and the provided



(a) time domain analysis

(b) frequency domain analysis

Figure 3. Representative post-flight analysis using short period model

sensitivity equations were used in the other cases, denoted "analytical." In cases 2 and 6, the equations of motion were integrated to evolve the state vector, denoted "integration." In case 3, measurements of the state vector were substituted for the entire analysis, denoted "substitution," whereas in case 4 this was done for the first iteration only, denoted "switching." Results for substituting measured state histories in the frequency domain data are not shown because the estimates diverged.

Results for cases 1, 2, 5, and 6 show that computing output sensitivities using finite differences and the sensitivity equations yielded the same results and used the same number of iterations. However, using the sensitivity equations used about half as much computation time as finite differences. Case 3 used more iterations and more computation time than Case 2, and resulted in biased estimates. However, the results for case 4 show that substituting state histories for only the first iteration results in the same answer, but in fewer iterations and with less computation time. Cases 5 and 6 show that while more iterations were needed for convergence in the frequency domain, they were about twice as fast as using time domain data. The error bounds here are also lower.

Table 2. Results of output-error variations using simulation data

case	domain	sensitivity	$\hat{C}_{L_\alpha} \pm \hat{\sigma}$	$\hat{\theta}_7 \pm \hat{\sigma}$	time [s]	no. of	iteration
no.		calculation				iterations	time [s]
1	time	numerical / integration	$+4.943 \pm 0.177$	$+0.227 \pm 0.008$	2.94	21	0.140
2	time	analytical / integration	$+4.943 \pm 0.177$	$+0.227 \pm 0.008$	1.81	21	0.086
3	time	analytical / substitution	$+5.024 \pm 0.192$	$+0.237 \pm 0.009$	2.18	26	0.084
4	time	analytical / switching	$+4.943 \pm 0.177$	$+0.227 \pm 0.008$	1.66	19	0.087
5	freq	numerical / integration	$+4.958 \pm 0.061$	–	7.60	90	0.084
6	freq	analytical / integration	$+4.958 \pm 0.061$	–	3.27	90	0.036

To further illustrate these results, a typical history of the C_{L_α} estimate is shown in Figure 4 for cases 1–4. Uncertainties are not shown, for clarity. Cases 1–2 follow the same path and converge on the same solution. In case 3, the solution quickly approaches the true answer, but overshoots and asymptotes. Case 4 initially follows the path of case 3, but then switches to evaluating the sensitivity equations, which allows it to converge on the same answer as in cases 1–2, but faster and with fewer iterations.

Based on these results, the following are recommended for real-time implementation of output error:

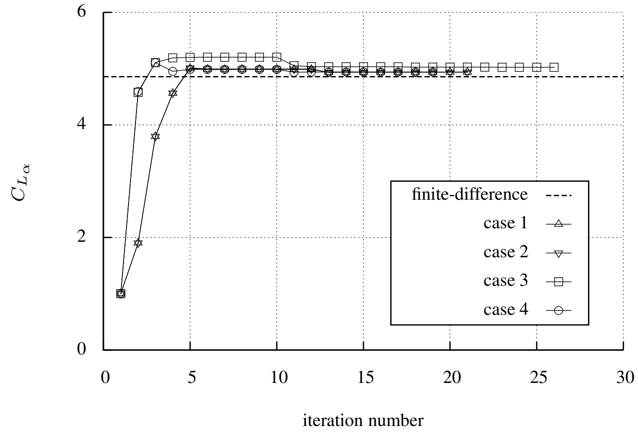


Figure 4. Comparison of estimated short period parameter C_{L_α} evolution

- Use frequency domain data unless estimating bias terms or the dynamics are nonlinear
- If practical, provide the sensitivity equations
- If sensitivity equations are provided, state vector measured or reconstructed, and time domain data is used, employ Denery's method for the first iteration to expedite convergence

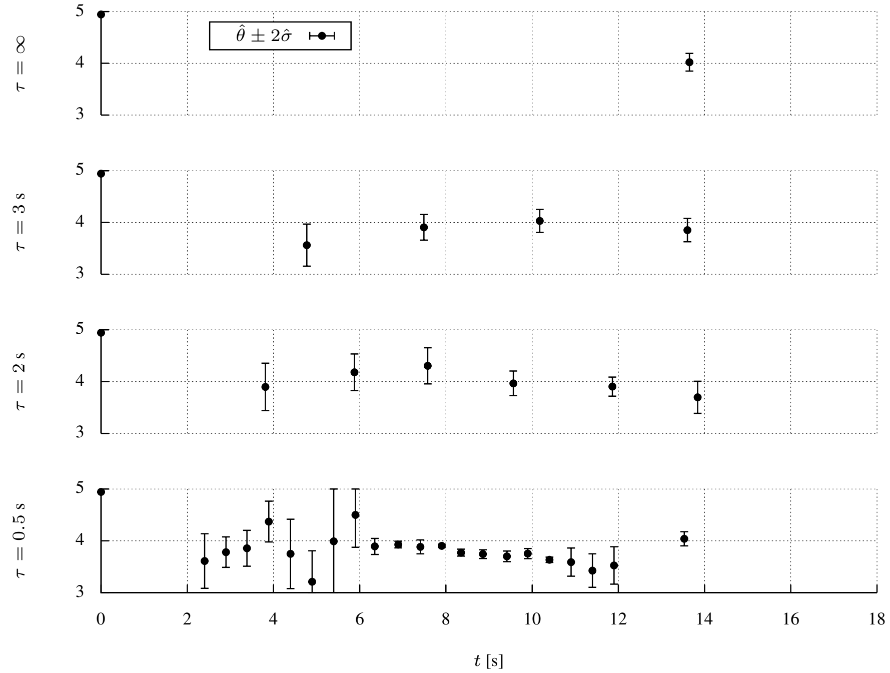
B. Real-Time Estimation of Short Period Mode

In this section, output error is applied to flight data in a real-time manner to estimate the short period model. This was done in both the time and frequency domains, using the recommendations of the previous section. The GTM simulation was used to compute starting values for the model parameters using numerical linearization. The sensitivity equations were provided, and Denery's method was used for the first iteration with time domain data. Corrections were applied to uncertainty estimates when using time domain data.

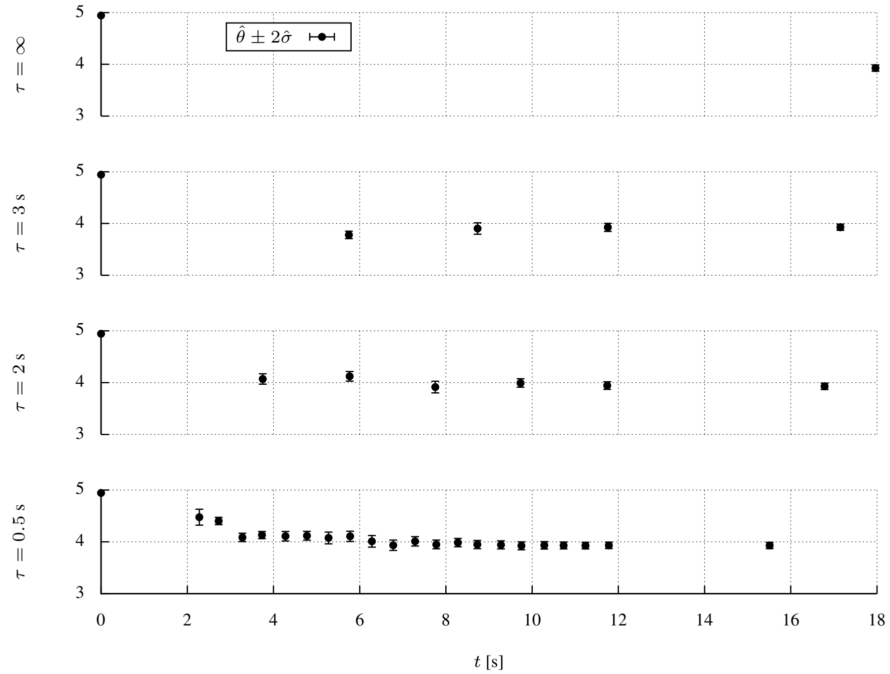
The estimation results using time domain data are shown in Figure 5(a) for four different update rates. For clarity, only the C_{L_α} estimate is shown. The first plot shows the post-flight solution, which was initiated after all the data were recorded and used 2.4 s of computation for convergence. In the second plot, updates are requested after collecting 3 s of data. In each analysis, the solver has enough time and information to converge on a result. The results evolve and the error bounds decrease as more information is retained, ultimately settling on a solution that is in statistical agreement with the batch solution. In the third plot, updates are requested every 2 s. The algorithm was interrupted in the majority of these analyses, although the estimates were still relatively close and in statistical agreement with the batch solution. In the fourth plot, updates were requested every 0.5 s. A 2 s initial delay was afforded because dynamic data is needed for parameter estimation. In this case, the output-error algorithm was interrupted during each analysis, and outputs were not able to converge. This led to sporadic parameter estimates having large error bounds. The algorithm had enough information to converge for the final estimate at the end of the data collection.

A similar analysis, shown in Figure 5(b), was repeated using frequency domain data. The first plot shows the batch solution for the C_{L_α} estimate. The estimate is in statistical agreement with the time domain batch analysis, but has much smaller error bounds and required 6.1 s of computation for convergence. The following three plots show results for when updates are requested every 3 s, 2 s, and 0.5 s, respectively. Because the frequency domain analysis requires more time for convergence than the time domain analysis, the algorithm is interrupted for every analysis. However, the error bounds are relatively small and the parameter estimates never stray far from the batch solution, unlike in the time domain. This is because the Fourier transforms are cumulative, so that the parameter estimates continue to improve as time goes on.

In summary, short period aerodynamic model terms can be estimated in real time using output error. Accurate results using time domain data are available every 2 s, while results using frequency domain data are ready after 3 s of data. If more complicated models with additional parameters are to be estimated, these rates could decrease, and of course the update rates depend on the computer being used.



(a) time domain analysis



(b) frequency domain analysis

Figure 5. Real-time estimates of short period model parameter C_{L_α}

C. Real-Time Data Compatibility Analysis

Aircraft sensors contain errors that create kinematically inconsistent sets of data, which then bias aerodynamic modeling results. Figure 6 shows flight test measurements, as well as reconstructed values computed using the kinematic equations. The mismatch is due to bias and scale factor errors in the measurements. Correcting measurements for these errors is called the data compatibility problem, and can be solved using output error.[5]

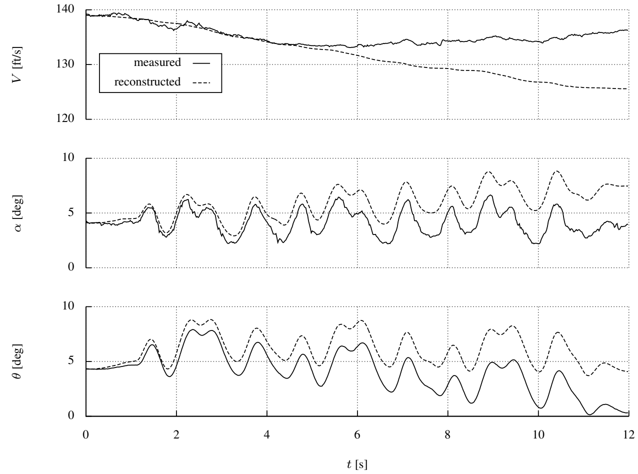


Figure 6. Measured and reconstructed longitudinal flight variables

The sensor measurements are modeled as

$$z(i) = (1+\lambda)y(i) + b + \nu(i) \tag{23}$$

where λ is a scale factor and b is a bias. For the data compatibility problem, the model parameters are

$$\boldsymbol{\theta} = \left[\begin{array}{cccccccccccccccccc} b_{a_x} & b_{a_y} & b_{a_z} & b_p & b_q & b_r & \lambda_V & \lambda_\beta & \lambda_\alpha & b_V & b_\beta & b_\alpha & \lambda_\phi & \lambda_\theta & \lambda_\psi & b_\phi & b_\theta & b_\psi \end{array}\right]^T \tag{24}$$

although it is common to use only a subset of these. The state equations are found by substituting the accelerometer and gyroscope measurements into the rigid body aircraft equations of motion as[5]

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & r-b_r & b_q-q \\ b_r-r & 0 & p-b_p \\ q-b_q & b_p-p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} -g\sin\theta + ga_x - b_{a_x} \\ g\sin\phi\cos\theta + ga_y - b_{a_y} \\ g\cos\phi\cos\theta + ga_z - b_{a_z} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p-b_p \\ q-b_q \\ r-b_r \end{bmatrix} \tag{25}$$

and the model output equations are[5]

$$\begin{aligned} V &= (1+\lambda_V)\sqrt{u^2+v^2+w^2} + b_V \\ \beta &= (1+\lambda_\beta)\arcsin\left(v/\sqrt{u^2+v^2+w^2}\right) + b_\beta \\ \alpha &= (1+\lambda_\alpha)\arctan\left(w/u\right) + b_\alpha \\ \phi &= (1+\lambda_\phi)\phi + b_\phi \\ \theta &= (1+\lambda_\theta)\theta + b_\theta \\ \psi &= (1+\lambda_\psi)\psi + b_\psi \end{aligned} \tag{26}$$

Control surface deflections are not included in this analysis because there is no redundant kinematic information for that data.

Data compatibility is a difficult estimation problem. The state equations are nonlinear and require computationally expensive and iteration solvers. Good starting values are needed and the algorithm can converge on incorrect solutions due to low sensitivities. Sufficient excitation is required to accurately estimate the model parameters, e.g. the airspeed scale factor would be difficult to estimate using data from a short-period maneuver, where airspeed is practically constant.

One approach to solving the data compatibility problem in real time is to linearize the system about the trim condition. To illustrate, applying partial differentiation to Eqs. 25 and 26, evaluating at the trim conditions for straight and level flight, and specializing to the longitudinal dynamics yields the linear system

$$\begin{aligned}
\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{w} \\ \Delta\dot{\theta} \end{bmatrix} &= \begin{bmatrix} 0 & -b_q & -g\cos\theta_0 \\ b_q & 0 & -g\cos\phi_0\sin\theta_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} 1 & 0 & -w_0 \\ 0 & 1 & u_0 \\ 0 & 0 & \cos\phi_0 \end{bmatrix} \begin{bmatrix} \Delta a_x \\ \Delta a_z \\ \Delta q \end{bmatrix} \\
\begin{bmatrix} \Delta V \\ \Delta\alpha \\ \Delta\theta \end{bmatrix} &= \begin{bmatrix} u_0(1+\lambda_V)/V_0 & w_0(1+\lambda_V)/V_0 & 0 \\ -w_0(1+\lambda_\alpha)/V_0^2 & u_0(1+\lambda_\alpha)/V_0^2 & 0 \\ 0 & 0 & 1+b_\theta \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta\theta \end{bmatrix}
\end{aligned} \tag{27}$$

Although the scale factors and gyroscope biases can be estimated using this system, the other biases do not appear in the equations and cannot be estimated. If data corrected using these equations was then used for aerodynamic parameter estimation, only the stability and control derivative estimates would be free of data compatibility related errors.

Another approach is to use the full nonlinear data compatibility equations in Eqs. 25 and 26. A time-domain analysis is required because the equations are nonlinear, but every unknown parameter can be estimated. In this case, the full data set can be corrected for data compatibility errors for further analysis. This approach is presented next, using flight data from the T-2 aircraft. The output sensitivity equations, derived by hand and checked using a computer algebra software called Maxima,[26] were supplied to expedite the estimation. Denery's method for state substitution was used for the first iteration to improve the convergence. Only the longitudinal equations were used in this case. Biases for the accelerometer and gyroscope, and scale factors for the angle of attack and pitch angle were estimated, which is typical.

Figure 7 shows the model parameters, estimated every 4 s. The parameters and uncertainties eventually settle with each segment of data. The final model parameter estimates are small because of high quality measurements and calm air during the flight test. The sensor measurements were corrected for each segment and are shown in Figure 8. Except for the longitudinal acceleration signal, these traces are almost indistinguishable. Estimation of the short period model parameters using this corrected data resulted in less than a 1% change in stability and control parameters and no change in the estimated uncertainties.

The 4 s update period was found to be a good trade off between accurate results and the update rate of the model parameters. The largest computational burden was the integration of the nonlinear equations and the uncertainty correction for the colored residuals. One option would be to check the whiteness of the residuals and skip this correction, because the uncertainties are only of secondary importance here, and because the kinematic equations are well known and can only be corrupted by process noise and unmodeled sensor errors. In this case, accurate model parameters could be estimated up to every 0.5 s.

V. Conclusions

Output error is a standard method for estimating unknown parameters in dynamic models of aircraft in post-flight analysis. This is due to a sound theoretical underpinning, favorable asymptotic properties, realistic assumptions consistent with the real flight test situation, and a straightforward approach that has been tested and improved upon using numerous flight test cases. However, there are drawbacks that have prevented its application to real-time parameter estimation during flight.

This paper addresses some of those difficulties by exploring variations of the traditional output-error approach that make it more suitable for real-time applications. Real-time estimation is becoming increasingly important because it helps to reduce time and costs associated with identifying dynamic models from flight test data. For instance, immediate modeling feedback can shorten maneuver time durations or help diagnose problems while the aircraft is still on condition, instead of after the flight test.

Experiments using simulation data to estimate short period model parameters led to several suggestions for real-time implementation. Frequency domain data should be used unless the dynamics of interest are

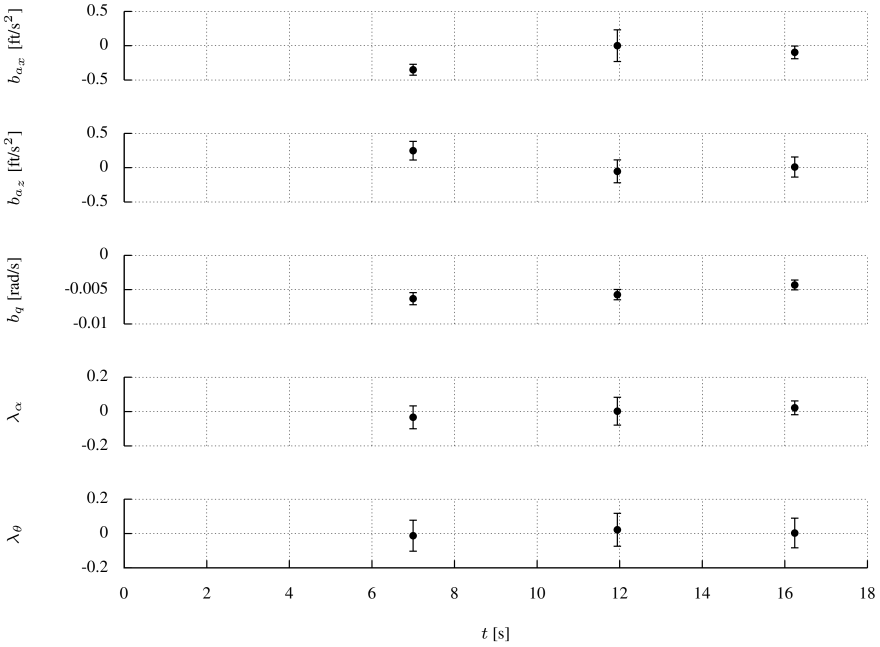


Figure 7. Real-time estimates of data compatibility parameters

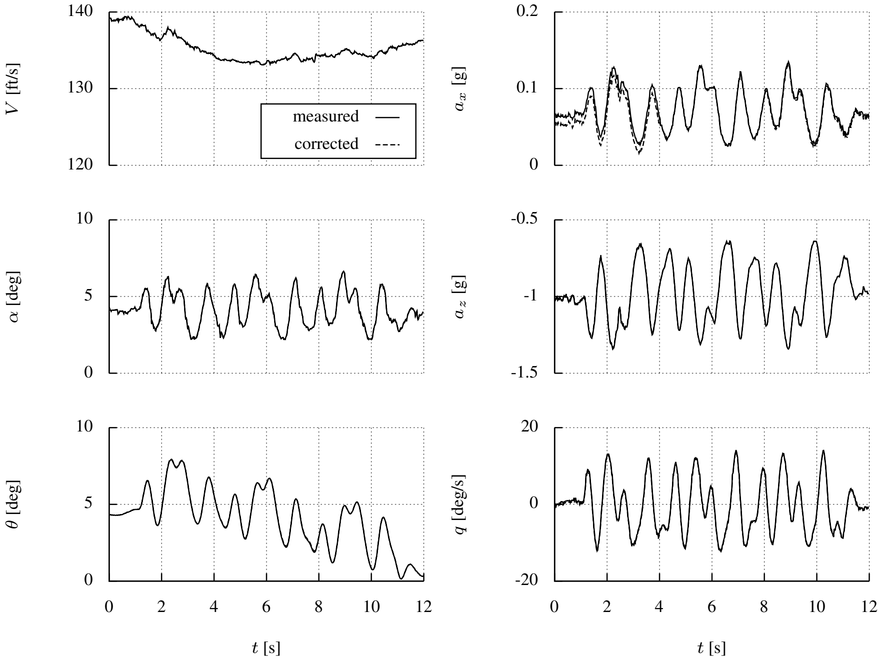


Figure 8. Sensor measurements corrected for estimated biases and scale factors

nonlinear and cannot be linearized, or if bias terms are of interest. If practical, the output sensitivity equations should be derived and provided to the estimator to increase computation speed. In this case, Denery's method can be used with time-domain data to expedite convergence from arbitrary starting values for the model parameters. Application to flight test data using MATLAB®code and a laptop computer demonstrated that short period model estimates could be estimated every 2 s in the time domain, or more rapidly in the frequency domain after 3 s of data. A list of recommendations for increasing computational speed was provided. Model size (number of states, outputs, and parameters) as well as the capability of the computing equipment affect the achievable update rate.

Application to flight test data also showed that the data compatibility problem could be solved every 4 s, or as fast as every 0.5 s if the error bound corrections for colored residuals are skipped. Data compatibility is an important and standard step in system identification that ensures a kinematically consistent set of data, prior to aerodynamic modeling. In the case presented, the corrections were small and only led to a 1% change in the estimated short period estimates. However, in other cases this is not the case and corrections could be much larger. A 4 s update rate may be slower than desired because the aerodynamic estimation can update faster, but again the uncertainty correction can be skipped and the code run time can be shortened. In addition to correcting data for aerodynamic modeling, solving the data compatibility problem in real time can be used for flight operations data checks or for fault detection. If estimated bias or scale factors are abnormally large, the flight test engineers can be alerted to a potential fault in the associated measurements.

There still remain shortcomings of applying output error in real time. A model structure is needed before modeling; however in many instances, such as perturbation maneuvers, low-order equivalent system modeling, and data compatibility analysis, an adequate model structure can be assumed. Due to assumptions, output error is still not equipped to deal with the effects of turbulence and other state disturbances. Bias parameters can only be estimated using time domain data because of detrending prior to transforming the data into the frequency domain. Output error still is fundamentally an iterative technique that requires more time than non-iterative methods. Finally, a shortcoming of Denery's method using arbitrary starting values of the model parameter estimates was identified when using frequency domain data.

This method provides another approach to real-time dynamic modeling as the aircraft is flying. Bias parameters, such as the profile drag, can be estimated directly in real time. The data compatibility analysis will provide flight test engineers with real-time updates to sensor measurements, and the ability to check for proper sensor function in real time. The data compatibility analysis could be integrated into future algorithms for fault detection. This method could also be used for other modeling applications, such as fitting transfer functions to real-time frequency response data. Data forgetting techniques could also be incorporated to make the algorithm more rapidly adapt to changing dynamics.

VI. Acknowledgments

This research was funded by the NASA Aviation Safety Program, Vehicle Systems Safety Technologies project, and the Subsonic Fixed-Wing Project. Conversations with Eugene Morelli at NASA Langley Research Center are acknowledged and appreciated.

References

[1] Taylor, L. and Iliff, K., "System Identification Using a Modified Newton-Raphson Method – A FORTRAN Program," Tech. Rep. TN–D–6734, NASA, Washington, D.C., May 1972.

[2] Maine, R. and Iliff, K., "A FORTRAN Program for Determining Aircraft Stability and Control Derivatives From Flight Data," Tech. Rep. TN–D–7831, NASA, Edwards, CA, April 1975.

[3] Maine, R. and Iliff, K., "User's Manual for MMLE3, a General FORTRAN Program for Maximum Likelihood Parameter Estimation," Tech. Rep. Technical Paper 1563, NASA, Edwards, CA, November 1980.

[4] Murray, J. and Maine, R., "pEst Version 2.1 User's Manual," Tech. Rep. TM–88280, NASA, Edwards, CA, September 1987.

[5] Klein, V. and Morelli, E., *Aircraft System Identification: Theory and Practice*, AIAA Education Series, AIAA, 2006.

[6] Ranaudo, R., Martos, B., Norton, B., Gringas, D., Barnhart, B., Ratvasky, T., and Morelli, E., "Piloted Simulation to Evaluate the Utility of a Real Time Envelope Protection System for Mitigating In-Flight Icing Hazards," No. 2010–7987 in Atmospheric and Space Environments conference, AIAA, Toronto, Ontario, Canada, August 2010.

[7] Morelli, E. and Klein, V., "Determining the Accuracy of Maximum Likelihood Parameter Estimates With Colored Residuals," Tech. Rep. CR–194893, NASA, Hampton, VA, September 1994.

[8] Morelli, E. and Klein, V., "Accuracy of Aerodynamic Model Parameters Estimated From Flight Test Data," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, January–February 1997, pp. 74–80.

[9] Maine, R. and Murray, J., "Application of Parameter Estimation to Highly Unstable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 3, May-June 1988, pp. 213–219.

[10] Hui, K., Auriti, L., , and Ricciardi, J., "Advances in Real-Time Aerodynamic Model Identification," *Journal of Aircraft*, Vol. 42, No. 1, January-February 2005, pp. 73–79.

[11] Jategaonkar, R. and Plaetschke, E., "Estimation of Aircraft Parameters Using Filter Error Methods and Extended Kalman Filter," Forschungsbericht FB 88-15, DFVLR, 1988.

[12] Garcia-Velo, J. and Walker, B., "Aerodynamic Parameter Estimation for High-Performance Aircraft Using Extended Kalman Filter," No. 95–3500, AIAA, 1995.

[13] Morelli, E., "Real-Time Parameter Estimation in the Frequency Domain," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, Sept.–Oct. 2000, pp. 812–818.

[14] Grauer, J. and Morelli, E., "Method for Real-Time Frequency Response and Uncertainty Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, January–February 2014, pp. 336–343.

[15] Maine, R. and Illif, K., "Application of Parameter Estimation to Aircraft Stability and Control: The Output-Error Approach," Tech. Rep. 1168, NASA, June June.

[16] Klein, V., "Maximum Likelihood Method for Estimating Airplane Stability and Control Parameters From Flight Data in Frequency Domain," Technical Paper 1637, NASA, Hampton, VA, May 1980.

[17] Morelli, E., "High Accuracy Evaluation of the Finite Fourier Transform Using Sampled Data," Tech. Rep. TM–110340, NASA, Hampton, VA, June 1997.

[18] Denery, D., "An Identification Algorithm That Is Insensitive to Initial Parameter Estimates," *AIAA Journal*, Vol. 9, No. 3, March 1971, pp. 371–377.

[19] Jordan, T., Foster, J., Bailey, R., and Belcastro, C., "AirSTAR: A UAV Platform for Flight Dynamics and Control System Testing," No. 2006-3307 in Aerodynamic Measurement Technology and Ground Testing conference, AIAA, San Francisco, CA, June 2006.

[20] Jordan, T. and Bailey, R., "NASA Langley's AirSTAR Testbed – A Subscale Flight Test Capability for Flight Dynamics and Control System Experiments," No. 2008-6660 in Guidance, Navigation, and Controls conference, AIAA, Honolulu, HI, Aug. 2008.

[21] Shah, G., Cunningham, K., Foster, J., Fremaux, M., Stuart, E., Wilborn, J., Gato, W., and Pratt, D., "Wind-Tunnel Investigation of Commercial Transport Aircraft Aerodynamics at Extreme Flight Conditions," No. 2002–01–2912 in World Aviation Congress and Display, SAE, Phoenix, AZ, November 2002.

[22] Morelli, E. and DeLoach, R., "Wind Tunnel Database Development using Modern Experiment Design and Multivariate Orthgonal Functions," No. 2003–653 in 41st AIAA Aerospace Sciences Meeting and Exhibit, AIAA, Reno, NV, January 2003.

[23] Morelli, E., "Flight-Test Experiment Design for Characterizing Stability and Control of Hypersonic Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, May–June 2009, pp. 949–959.

[24] Morelli, E., "Flight Test Maneuvers for Efficient Aerodynamic Modeling," No. 2011-6672 in Atmospheric Flight Mechanics conference, AIAA, Portland, OR, Aug. 2011.

[25] MATLAB, "Version 8.3.0.532 (R2014a)," 2014.

[26] Maxima, "Maxima, a Computer Algebra System. Version 5.25.1," 2011.