

Online Control Design for Learn-to-Fly

Steven Snyder*, Barton Bacon†, and Eugene Morelli‡
NASA Langley Research Center, Hampton, VA, 23681

Susan Frost§, Christopher Teubert¶, and Wendy Okolo||
NASA Ames Research Center, Moffett Field, CA, 94035

Two methods were developed for online control design as part of a flight test effort to examine the feasibility of the NASA Learn-to-Fly concept. The methods use an aerodynamic model of the aircraft that is being identified in real-time onboard the aircraft to adjust the control parameters. One method employs adaptive nonlinear dynamic inversion, whereas the other consists of a classical autopilot structure. Effects from the interaction between the real-time modeling and the developed control laws are discussed. The Learn-to-Fly concept has been deemed feasible based on successful flights of both a stable and unstable aircraft.

I. Nomenclature

b	=	wing span
\bar{c}	=	wing mean aerodynamic chord
C_l, C_m, C_n	=	body-axis nondimensional aerodynamic moment coefficients
g	=	acceleration due to gravity
$I_{xx}, I_{yy}, I_{zz}, I_{xz}$	=	inertia tensor elements
J	=	advance ratio
K	=	control gain
L	=	aerodynamic lift
m	=	mass
M	=	vector of aerodynamic moments
M_δ	=	vector of aerodynamic moments due to control surface deflection
n_c	=	number of modeling regressors
N	=	number of data points
p, q, r	=	body-axis roll, pitch, and yaw rates
\bar{q}	=	dynamic pressure
R	=	propeller radius
S	=	wing reference area
T_x, T_z	=	body x -axis and body z -axis components of engine thrust
Th	=	throttle
V	=	true airspeed
V_{CAS}	=	calibrated airspeed
X	=	matrix of modeling regressors
Y	=	aerodynamic side force
z	=	vector of observations
α	=	angle of attack
β	=	sideslip angle
δ	=	surface deflection
ϵ	=	vector of residuals

*Research Engineer, Dynamic Systems and Controls Branch, MS 308

†Senior Research Engineer, Dynamic Systems and Controls Branch, MS 308, and Member AIAA

‡Senior Research Engineer, Dynamic Systems and Controls Branch, MS 308, and Associate Fellow AIAA

§Computer Engineer, Intelligent Systems Division, MS 269-3, and Associate Fellow AIAA

¶Computer Engineer, Intelligent Systems Division, MS 269-3, and Member AIAA

||Research Engineer, Intelligent Systems Division, MS 269-3, and Member AIAA

ζ	=	damping ratio
θ	=	vector of modeling parameters
μ, γ, χ	=	bank, flight path, and track angles
ϕ, θ	=	Euler roll and pitch angles
ω	=	angular rate of the body with respect to the inertial frame
ω_n	=	natural frequency
Ω	=	propeller speed

Superscripts

\top	=	matrix transpose
-1	=	matrix inverse
\cdot	=	time derivative
$\hat{}$	=	estimate
\sim	=	error

Subscripts

cmd	=	command
lf, rf	=	left and right flap
la, ra	=	left and right aileron
ev	=	elevator
ru	=	rudder

II. Introduction

FOR decades, the development of flight control laws has been dominated by a paradigm built on ground-based modeling of aerodynamics, relying on wind-tunnel testing or computational fluid dynamics. A new paradigm, Learn-to-Fly (L2F), has been introduced [1] and incorporates real-time modeling [2], real-time guidance [3], and learning control in order to develop an aerodynamic model and control law online. The work presented here is part of a feasibility study of the Learn-to-Fly concept. While other modeling and control techniques could also be employed [4, 5], this paper examines two approaches to the Learn-to-Fly online control design problem. One approach is based on a nonlinear dynamic inversion method with the addition of an adaptive disturbance rejection module. The other uses a classical autopilot structure.

The idea of combining real-time model identification with online control design has been considered before in addressing the problem of reconfigurable flight controls. Reconfigurable flight control systems must adapt to unknown failures and damage. Self-designing control that accommodates failures has successfully been flown on the VISTA/F-16 using regularized sequential least-squares parameter estimation to estimate the stability and control derivatives required for the solution of a receding-horizon optimal control problem [6]. Significant research has been performed in areas of system identification, control allocation, and nonlinear dynamic inversion to formulate reconfigurable controls. Chandler [7, 8] developed a system identification scheme which uses a priori information to enhance parameter estimates, based on least-squares estimation. Abrupt changes in aircraft dynamics due to failure are accurately modeled with windowing techniques. Persistent excitation is required, however, to prevent erroneous parameter estimation. A variety of approaches to control allocation have been proposed for use with reconfigurable systems [9, 10]. Buffington proposed a control allocation approach that prevents actuator saturation and yields a graceful degradation for any unachievable commands in dynamic inversion control laws [11]. Wise et al. [12] proposed a reconfiguration flight control law based on dynamic inversion that utilizes an on-line neural network, derived from the work of Kim and Calise [13], to adaptively regulate the error in plant inversion. The network stabilizes the system following failure and serves to alleviate the criticality of the system identification. The online control allocation generates commands to yield angular accelerations and alleviate loading during maneuvers.

All of these approaches to reconfigurable controls have a great deal of information about the vehicle prior to failure. Any viable Learn-to-Fly controller must initially control the vehicle with scant information about its aerodynamics. Later, the control modifies itself based on the aerodynamic model that is concurrently being identified. One of the challenges faced by the overall system is that a control law that is good at tracking typically rejects disturbances. However, modeling injects small disturbance inputs in order to excite the vehicle response for identification. Attenuation of these disturbances by the control law makes it difficult to identify an accurate model. Moreover, making appropriate modifications to the control law requires some level of model fidelity. In this way, the identification and control objectives are at odds with each other.

For the first approach to the Learn-to-Fly problem, an adaptive nonlinear dynamic inversion controller is proposed that replaces the modeled aerodynamics with a set of desired dynamics computed from the identified model, a global model of the vehicle's non-dimensional forces and moments. From this model, key stability and control derivatives are available for formulating control. Unlike the work cited above, the online modeling can function with little excitation. Due to its heavy dependence on modeling output, concurrent nonlinear dynamic inversion control could produce undesirable model-control coupling. One of the control variables, angle of attack, was also one of the principal explanatory variables for modeling, potentially compromising the effort to produce good models. Regardless, the adaptive nonlinear dynamic inversion controller proposed was able to achieve high performance at the expense of this increased model dependency.

In the second approach, a classical autopilot design is proposed that seeks a moderate level of control to intentionally avoid adverse coupling with modeling. Low-order linear models from modeling were used to adjust a few stability augmentation gains in the inner loop of the classical autopilot. The control variables were primarily selected to be the vehicle's attitude, not the explanatory variables of the aerodynamic model being constructed. The control generated commands for a pseudo set of standard aircraft surfaces. Using additional modeling information, the primary pseudo surface commands from the autopilot were interpreted as moment commands that could be sent to the control allocator. Reasonable performance was achieved in the simulation with the classically-based control.

In this paper, the key learning feature of the L2F system is discussed along with issues arising in performing concurrent identification with online control design. The term modeling used in this paper refers to real-time global aerodynamic modeling [2] and is summarized in the next section. The control algorithms for both the adaptive nonlinear dynamic inversion control and the classically-based control are presented. Flight results of both approaches are discussed.

III. Learning

The difference between adaptive control and learning control is memory [14]. Adaptive control has no memory whereas a learning control both adapts and has memory. Learning control action is based on past experience with the environment.

The L2F algorithm memory lies exclusively in the real-time modeling algorithm that compactly retains all the information of the past to construct models for the aircraft nondimensional force and moment coefficients. The control utilizes aerodynamic modeling information to set reference dynamics, adjust gains, inject inverse dynamic inputs, and allocate use of control surfaces. This differs from reinforcement learning [15], where the prescribed control action springs directly from past experience in the memory of what works and does not work—a rather costly approach to flight control due to the high cost of experiencing failure. In the past, the control strategy used in this paper has been called indirect-adaptive control [16].

To be able to discuss some of the issues that can arise when simultaneously identifying an aerodynamic model for the aircraft and using that information to control the aircraft, it is appropriate to summarize some of the important features of the real-time modeling algorithm found in [2].

The real-time modeling uses a time-domain equation-error algorithm that combines recursive regressor orthonormalization, least-squares parameter estimation, and statistical modeling metrics to select the best model terms for an accurate global aerodynamic model from a prescribed pool of candidate modeling terms. Underlying the algorithm is the proposition that aircraft nondimensional forces and moments can be accurately characterized by linear and nonlinear functions of a specified set of explanatory variables. These include angle of attack, sideslip angle, nondimensional body-axis angular rates, and control surface positions. Prior to any model building, measured data for the explanatory variables, along with accelerometer data used to construct forces experienced, are locally smoothed using real-time procedures described in [17]. The angular accelerations required to construct the moments are computed using a real-time smooth numerical differentiation of the body-axis angular rates. Both the smoothing technique and the smooth numerical differentiation incur a two-sample delay at 50 Hz for the data being used to identify real-time models.

For each component of nondimensional aerodynamic force and moment, a model is postulated as a linear regression, with model terms to be selected as a subset of a prescribed large comprehensive pool of model terms, also called regressors. For L2F real-time global aerodynamic modeling, these regressors were polynomial terms (linear, bilinear, squared, etc.) of the smoothed explanatory variables, but, in general, they can be arbitrary nonlinear functions of the smoothed explanatory variables. The regressors can also include absolute values of explanatory variables, which is useful for axial force contributions due to control surface deflections. The most important requirement, though, is that the regressor pool be comprehensive to facilitate accurate real-time global modeling. However, practical limitations

of the onboard computing capability restricts the size of the candidate regressor pool, and this, of course, affects the real-time global modeling capability and accuracy. The compact way that information is stored in memory (to be described), consisting of information pertaining to past regressor terms and past observations of nondimensional aerodynamic force and moment coefficients, makes it possible to select important regressors and identify accurate least-squares model fits for all of the data.

To briefly describe this compact form, consider first a non-recursive least-squares problem. Let \mathbf{z} denote an $N \times 1$ vector of observations and let \mathbf{X} denote an $N \times n_c$ matrix of modeling functions or regressors, where each column contains one of the regressors. The least-squares problem is to compute an $n_c \times 1$ vector of model parameters $\boldsymbol{\theta}$ that minimizes the sum of squares of the elements of the $N \times 1$ residual vector $\boldsymbol{\varepsilon}$ defined by

$$\boldsymbol{\varepsilon} = \mathbf{z} - \mathbf{X}\boldsymbol{\theta}.$$

The solution to this problem satisfies the normal equations

$$\mathbf{X}^\top \mathbf{X} \boldsymbol{\theta} = \mathbf{X}^\top \mathbf{z}.$$

To solve, a \mathbf{QR} decomposition of \mathbf{X} is used,

$$\mathbf{X} = \mathbf{Q}\mathbf{R},$$

where \mathbf{Q} is an $N \times n_c$ matrix whose columns are orthonormal and \mathbf{R} is an $n_c \times n_c$ upper triangular matrix. The normal equations become

$$\mathbf{R}^\top \mathbf{Q}^\top \mathbf{Q} \mathbf{R} \boldsymbol{\theta} = \mathbf{R}^\top \mathbf{Q}^\top \mathbf{z},$$

or

$$\mathbf{R} \boldsymbol{\theta} = \mathbf{Q}^\top \mathbf{z}.$$

Since \mathbf{R} is nonsingular if $\mathbf{X}^\top \mathbf{X}$ is nonsingular,

$$\boldsymbol{\theta} = \mathbf{R}^{-1} \mathbf{Q}^\top \mathbf{z}.$$

Gentleman [18, 19] showed that \mathbf{Q} is not actually needed to solve for $\boldsymbol{\theta}$. A series of row operations—applied to both \mathbf{X} and \mathbf{z} so that the sub-diagonal part of each column of $\begin{bmatrix} \mathbf{X} & \mathbf{z} \end{bmatrix}$ is zeroed—will produce both \mathbf{R} and $\mathbf{Q}^\top \mathbf{z}$ directly, in addition to the root residual sum of squares, $\sqrt{\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}}$, so that

$$\begin{bmatrix} \mathbf{X} & \mathbf{z} \end{bmatrix} \xrightarrow[\text{operations}]{\text{row}} \begin{bmatrix} \mathbf{R} & \mathbf{Q}^\top \mathbf{z} \\ 0 & \sqrt{\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^* \\ 0 \end{bmatrix}.$$

The elements of $\mathbf{Q}^\top \mathbf{z}$ are actually model parameters for the orthonormal basis functions defining y as the approximation of z ,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} = \mathbf{Q}(\mathbf{Q}^\top \mathbf{z}).$$

The model parameters $\boldsymbol{\theta}$ for the columns of \mathbf{X} are related to the model parameters for the columns of \mathbf{Q} by the \mathbf{R}^{-1} matrix. Consistent with the least-squares solution and the normal equations, the residual $\boldsymbol{\varepsilon} = \mathbf{z} - \mathbf{y}$ is always orthogonal to the columns of \mathbf{Q} .

More importantly, the matrix \mathbf{R}^* compactly stores all the information regarding past regressor and dependent variable data. The recursive form of the algorithm simply appends the next (i th) data for the regressors x_i and dependent variable z_i to \mathbf{R}^* as

$$\begin{bmatrix} \mathbf{R}^* \\ [x_i \quad z_i] \end{bmatrix}.$$

Row operations are then performed to zero the last row and create an updated \mathbf{R}^* .

At any point during the modeling process, statistically insignificant regressors can be removed by eliminating their respective columns from \mathbf{R}^* [18]. Row operations are then performed to make the column-truncated \mathbf{R}^* upper triangular, exploiting the existing zeros, to form \mathbf{R}_m^* . The model that includes all candidate model terms is defined by \mathbf{R}^* and the square of $\sqrt{\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}}$ is the residual sum of squares corresponding to including all candidate regressors from the \mathbf{X} matrix in the model, using all of the data. For model parameter uncertainty calculations, the L2F real-time modeling algorithm adjusts the residual sum of squares to account for candidate model terms that are not selected for inclusion in the global model. Note that it is costly in computer memory and computations to add new regressors and preserve a least-squares model fit for all of the data. This would actually require saving all previous row operations and applying them to the new regressor data assembled from all previous explanatory variable data.

The matrix \mathbf{R}^* is not discarded, but rather used to process the next set of regressor and dependent variable data, and to incorporate their information in memory. There is a wealth of statistical information contained in \mathbf{R}^* ([18, 20]) that determines the covariance of the model parameters and enables significance t -tests and F -tests on individual regressors, as well as multiple correlation coefficients (R^2) and prediction error tests (PSE) on groups of regressors [2, 17].

In the L2F real-time modeling algorithm, \mathbf{R}^* is updated at 50 Hz; statistically significant model terms are identified (i.e. an \mathbf{R}_m^* is formed, which is the model structure determination) and the associated model parameters and uncertainties are computed at 5 Hz. Global models of the nondimensional force and moment coefficients, in terms of the linear and nonlinear regressors, are broadcast to the other components of the L2F algorithm, including real-time guidance and control. Analytic partial derivatives of the identified global models with respect to the explanatory variables, evaluated at desired conditions (updated at 50 Hz), provide the necessary stability and control derivatives for setting gains in the control laws.

From a controls perspective, certain conditions have to be met in order to expect reasonable models. The range of variation of the explanatory variables used to assemble the regressors must cover the intended range of operation, otherwise the results obtained from using the identified models will be extrapolations, which are known to have degraded fidelity. For excitation of the control surfaces, the L2F algorithm applies an orthogonal set of Programmed Test Inputs (PTIs), added directly to the surface commands from the controller. These, of course, also excite the body-axis angular rates and the airflow angles, which is their purpose.

Slowly varying commands for nominal angle of attack and sideslip angle over a large portion of the flight envelope are needed to generate a global model valid over a large range of flight conditions. A sufficiently high signal-to-noise ratio for the aircraft responses, as captured in the translational and rotational acceleration data, must be present for accurate modeling. This creates a paradox. Good control tends to mitigate disturbances, such as those due to PTIs. However, real-time modeling needs to observe the response due to the PTIs in order to accurately construct global aerodynamic models.

The real-time global modeling is based on a least-squares formulation that considers all data. Model terms and parameter values are determined that minimize the sum of squares of all residuals, past and present. One implication is that flight conditions visited more often will naturally get more emphasis in the modeling approximation, because of an increased number of data points. This can lead to degraded model predictions when extrapolating for flight conditions beyond those frequently encountered, particularly when using a model containing higher-order terms. The L2F real-time modeling was designed to work only when PTIs were active, simply because the resulting flight data with PTIs are more informative and less correlated, with higher signal-to-noise ratios compared to ordinary maneuvering flight. When PTIs stopped, the identified global models were frozen, but the local stability and control derivatives were still updated at 50 Hz using current flight information.

The L2F project encountered an adverse model-control problem in the development of the L2F algorithm for the Woodstock glider, a joined-winged glider with an articulated tail, eight surfaces distributed across the two wings, and two tip surfaces. The problem manifested itself as an intermittent high-frequency buzz on one or more surfaces of Woodstock when the modeling and control were concurrently operating. In the early development, the real-time modeling algorithm development and online control design proceeded separately. Development and testing for the real-time modeling algorithm was done using a simple control that directly drove the primary control surfaces to enable wings level gliding flight at a constant pitch angle. To determine the fidelity of the real-time modeling results, a polynomial model of the simulation aerodynamic database was used in the simulation. The polynomial model contained terms identical to the aerodynamic modeling regressors, as well as additional terms. The modeling algorithm readily identified a global model in real time for the polynomial model used in the simulation. Using the simplified control, the real-time modeling algorithm also performed well on the simulation database in straight gliding flight. The simulation database did not use polynomial terms, but instead was composed of data tables with no specific function form.

The controls group duplicated the simulation aerodynamic database in the controls and control allocator and added

partial derivative tables to construct the corresponding stability and control derivatives from the simulation. During the development and testing of the control laws, these tables served as a substitute for real-time modeling results and were used to generate simulation-defined modeling data to check out the adaptive control algorithm. This was the equivalent of perfect modeling information all of the time. Gains and delays were added to the simulation-defined modeling results to get a sense of the robustness of the algorithm. The control algorithm performed well when used alone with this source of modeling information.

As stated, such was not the case when Woodstock modeling and control algorithms operated concurrently. Several possible causes were investigated including the smoothness of the linearly interpolated aerodynamic database. Smoothing the database with splines and additional breakpoints produced no improvement. Noisy explanatory variables were suspect, but these suspicions were eliminated with filtering. A comparison of simulation-defined control derivatives and those obtained from the real-time aerodynamic modeling algorithm revealed rather large discrepancies of 30% and, in cases, 50% in the tip surface and the elevator contribution of the articulated tail. The discrepancy was attributed to lower-than-desired signal-to-noise ratios and attenuation of the PTIs in the surface activity with the controller active. The control was relaxed to make the disturbance rejection less aggressive and to make the signal-to-noise ratios higher, but still the problem persisted. The source of this problem was ultimately undetermined, prompting the development of an alternative controller (see Section IV.E).

This second control approach was based on the simplified one used for the real-time modeling algorithm development, but with additional capabilities. Specifically, it could perform controlled maneuvers, use the allocator to drive the surfaces, and perform limited augmentation by modifying the effective aerodynamic damping terms. Interestingly, this controller did not experience the coupling described for the adaptive controller, but it did experience the same modeling discrepancies in the predicted control derivatives when maneuvering.

Further investigation and an update to the modeling algorithm revealed that modifications to the candidate regressor pool solved the adverse model-control coupling problem. These modifications included removal of a cubic angle-of-attack term to avoid the extrapolation problem, adding a squared sideslip angle term to produce a symmetric contribution, and adding some bilinear terms containing products consisting of angle of attack and some control surface deflections. This underscores the importance of having a comprehensive candidate regressor pool for effective and accurate global aerodynamic modeling.

IV. Control Design

Given the real-time aerodynamic modeling information, the controller must determine actions that will enable the vehicle to follow commands provided by the guidance module. The guidance module determines the direction of the velocity vector in inertial space and provides track angle χ and flight path angle γ . For learning, the modeling system requires variations in angle of attack α and sideslip angle β . The guidance module constructs commands of these variables and passes them on to the control. Ideally, the controller must track χ , β , and either α or γ , depending on the mode. Of the two controllers considered, only the adaptive nonlinear dynamic inversion satisfied this objective. For reasons to be discussed, the classical control did not track α , but used variations in commanded pitch angle to indirectly enforce variations of angle-of-attack. Both controllers, however, utilized the same outer-loop control that translated flight path angle and track angle, respectively, to appropriate commands on pitch angle θ and roll angle ϕ . Both controllers used the same auto-throttle control for the powered vehicle.

To discuss the development of these controls, it is assumed that the vehicle dynamics are governed by the standard rigid body equations of motion [21]. Consider first these equations for unpowered flight:

$$\begin{aligned}
\dot{\chi} &= \frac{1}{mV \cos(\gamma)} (L \sin(\mu) + Y \cos(\mu) \cos(\beta)), \\
\dot{\gamma} &= \frac{1}{mV} (L \cos(\mu) - Y \sin(\mu) \cos(\beta) - mg \cos(\gamma)), \\
\dot{\phi} &= p + \tan(\theta) (q \sin(\phi) + r \cos(\phi)), \\
\dot{\beta} &= -\cos(\alpha)r + \sin(\alpha)p + \frac{1}{mV} (Y \cos(\beta) + mg \cos(\gamma) \sin(\mu)), \\
\dot{\alpha} &= q - \tan(\beta) (\cos(\alpha)p + \sin(\alpha)r) + \frac{1}{mV \cos(\beta)} (-L + mg \cos(\gamma) \cos(\mu)), \\
\dot{\theta} &= q \cos(\phi) - r \sin(\phi), \\
\dot{\omega} &= I^{-1} (M + M_\delta) - I^{-1} (\omega \times I \omega),
\end{aligned}$$

where $\omega = [p \ q \ r]^\top$ is the angular rate of the body with respect to the inertial frame, expressed in body coordinates, I is the inertia matrix, L and Y are the lift and side force, respectively, M is the external (aerodynamic) moment about the vehicle center of mass, and M_δ is the external moment due to the control surface deflections.

Under the following simplifying assumptions,

- 1) $\mu \approx \phi$,
- 2) $\cos(\alpha) \approx \cos(\gamma) \approx 1$,
- 3) $Y \approx 0$,
- 4) $\beta \approx 0$,
- 5) $L \cos(\phi) \approx mg$,
- 6) $\sin(\alpha)p$ negligible,

the dynamics become

$$\begin{aligned}
\dot{\chi} &= \frac{g}{V} \tan(\phi), \\
\dot{\gamma} &= 0, \\
\dot{\phi} &= p + \tan(\theta) (q \sin(\phi) + r \cos(\phi)), \\
\dot{\beta} &= -r + \frac{g}{V} \sin(\phi), \\
\dot{\alpha} &= q - \frac{g}{V} \sin(\phi) \tan(\phi), \\
\dot{\theta} &= q \cos(\phi) - r \sin(\phi), \\
\dot{\omega} &= I^{-1} (M + M_\delta) - I^{-1} (\omega \times I \omega).
\end{aligned} \tag{1}$$

A. Dynamic Inversion

To track the commands of each variable, nonlinear dynamic inversion (NDI) is sequentially employed to produce commands for the faster variables, exploiting the time-scale separation of the variables. The outer-most loop converts the χ and γ guidance commands to attitude commands ϕ and θ , respectively. For linear tracking, the desired derivatives are proportional to the error between the variable and its command. From the dynamics for χ in (1), this inversion process produces the command

$$\phi_{cmd} = \text{Sat}_{-45^\circ}^{+45^\circ} \left(\arctan \left(\frac{V}{g} K_\chi (\chi_{cmd} - \chi) \right) \right),$$

where the command has been additionally limited to lie within $\pm 45^\circ$. However, because $\dot{\gamma} = 0$ under the simplifying assumptions above, a faster variable cannot be used to achieve linear convergent dynamics for γ . Instead the following expression, resulting from a relation of the velocity vector's z -component in the inertial coordinate system, is directly solved for θ_{cmd} ,

$$\sin(\gamma_{cmd}) = a_1 \cos(\theta_{cmd}) + a_2 \sin(\theta_{cmd}),$$

where $a_1 = \cos(\alpha) \cos(\beta)$ and $a_2 = \sin(\phi) \sin(\beta) + \cos(\phi) \sin(\alpha) \cos(\beta)$ [22].

The next inversion step treats p , q , and r as inputs to the dynamics governing ϕ , α or θ , and β , respectively. Based on the simplified dynamics in (1),

$$\begin{aligned} p_{cmd} &= K_\phi(\phi_{cmd} - \phi) - \tan(\theta) (q \sin(\phi) + r \cos(\phi)), \\ q_{cmd} &= \begin{cases} K_\alpha(\alpha_{cmd} - \alpha) + \frac{g}{V} \sin(\phi) \tan(\phi), & \text{if in } \alpha \text{ mode} \\ \frac{1}{\cos(\phi)} (K_\theta(\theta_{cmd} - \theta) + r \sin(\phi)), & \text{if in } \theta \text{ mode} \end{cases} \\ r_{cmd} &= -K_\beta(\beta_{cmd} - \beta) - \frac{g}{V} \sin(\phi). \end{aligned}$$

Note that the q_{cmd} used is determined by which mode the vehicle is in, α -tracking mode or θ -tracking mode.

The fastest inner loop tracking these angular rate commands inverts their dynamics to produce a desired angular acceleration proportional to the angular rate error

$$M_{\delta,cmd} = IK_\omega(\omega_{cmd} - \omega) - (\hat{M} - \omega \times I\omega).$$

The $M_{\delta,cmd}$ equation contains the estimate of the current moment on the aircraft \hat{M} , which is determined by the current identified model of the aircraft. Determination of the proportionality constants K is discussed in Section IV.C.

B. Adaptive Disturbance Rejection

The adaptive disturbance rejection portion of the controller is designed to assist the inner control loop regulating the angular rates. It operates by comparing the measured response with an internal model of the desired dynamics. It then adjusts the control signal based on the difference between the desired and measured behaviors. The desired behavior is defined as the dynamic inversion component successfully canceling the nonlinearities and enforcing the linear response:

$$\dot{\omega}_{des}(t) = K_\omega(\omega_{cmd}(t) - \omega_{des}(t)).$$

It is known that the dynamic inversion will not be able to exactly achieve its goal, due to the simplifying assumptions, model uncertainty, external disturbances, etc. Writing the system dynamics in a form similar to the desired behavior and collecting everything else as a lumped input disturbance d ,

$$\dot{\omega}(t) = K_\omega(\omega_{cmd}(t) - \omega(t)) + u_{ad}(t) + d(t), \quad (2)$$

where u_{ad} is the scaled adaptive portion of the control input, $I^{-1}\Delta M$. An adaptive input will be defined below to reduce the impact of the disturbance.

The internal model of the desired dynamics is expressed as

$$\dot{\hat{\omega}}(t) = K_\omega(\omega_{cmd}(t) - \omega(t)) + u_{ad}(t) - K_{ad}\tilde{\omega}(t), \quad (3)$$

where $\tilde{\omega} \triangleq \hat{\omega} - \omega$. The last term in (3) acts as an observer term. Computing the error dynamics from (3) and (2),

$$\dot{\tilde{\omega}}(t) = -K_{ad}\tilde{\omega}(t) - d(t). \quad (4)$$

Expressing (4) in the Laplace domain, it becomes clear that the adaptive input u_{ad} can be made to cancel a low-pass version of the disturbance d by letting

$$u_{ad}(s) \triangleq K_{ad}\tilde{\omega}(s) = -\frac{K_{ad}}{s + K_{ad}}d(s). \quad (5)$$

This adaptive input can then be rescaled and added to the NDI moment command to generate the total moment command:

$$M_{\delta,cmd} = IK_{ad}\tilde{\omega} + IK_\omega(\omega_{cmd} - \omega) - (\hat{M} - \omega \times I\omega).$$

C. Determining Control Parameters

Since there were no specified performance requirements, a simple philosophy was adopted to determine the control parameters. First, maintain the frequency of the system, trying neither to speed up the response nor slow it down. Second, select an adequate damping for smooth tracking of the performance variables. Approximations were made to keep the inner workings simple and straightforward. The desired frequency was determined based on the aerodynamic derivatives of the global model linearized with all explanatory variables zero, except for the angle of attack, for which the commanded value was used. This is taken to be the design point, which will not, in general, be the same as linearizing the global model at the current operating point. The frequencies were calculated as shown in Table 1. Since the rolling moment is not dependent on the roll angle, the frequency was instead calculated using half of the moment due to aileron deflection. This leaves room for the aileron to be used for other tasks, such as canceling the nonlinearities or adjusting the damping. Alternatively, the location of the roll pole (using C_{l_p}), could be used to determine the frequency. The approach taken would depend on the vehicle and the given mission requirements.

It was assumed that the dynamic inversion commands were able to remove the nonlinear effects and only the desired linear portions remain. Looking at the short-period dynamics as an example and applying the commands as described above, the dynamics are

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_q K_\alpha & -K_q \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ K_q K_\alpha \end{bmatrix} \alpha_{cmd},$$

which is a canonical description of a second-order system. The control parameters can then be written in terms of the natural frequency and damping of the second-order system

$$\begin{aligned} K_q &= 2\zeta\omega_n \\ K_\alpha &= \omega_n/2\zeta. \end{aligned}$$

The desired frequency was determined based on the modeling results as in Table 1, and the desired damping ratio was selected to be 0.8. The control parameters for the roll channel (K_ϕ , K_p) and the yaw channel (K_β , K_r) are determined analogously. The control parameter for following the track angle, K_χ was fixed at 0.375. For the range of roll frequencies expected, the linear system set by applying the dynamic inversion would remain stable and provide adequate performance with this value fixed. This parameter could also be updated based on the modeling results but was not in this preliminary work.

D. Auto-Throttle

The auto-throttle managed the vehicle's calibrated air speed. A simple PI controller was implemented based on simulation data from a similar propeller-driven aircraft. The gains were fixed throughout the tests. No updates were made since an online estimate of a propulsion model was not available.

$$Th = K_p(V_{CAS,cmd} - V_{CAS}) + K_I \int_0^t (V_{CAS,cmd} - V_{CAS}) d\tau$$

To reduce transients when switching between the piloted mode and the autonomous mode, the autonomous command was added to the pilot's last throttle command. Thus, if the pilot had trimmed the aircraft at the appropriate speed before switching, the throttle command would remain the same. Additionally, the commands were limited to stay within the range of 0 to 1.

Table 1 Desired frequency calculations

Roll	$\omega_n = \sqrt{\left \frac{\bar{q} S b}{2 I_{xx}} C_{l\delta a} \right }$
Pitch	$\omega_n = \sqrt{\left \frac{\bar{q} S \bar{c}}{I_{yy}} C_{m\alpha} \right }$
Yaw	$\omega_n = \sqrt{\left \frac{\bar{q} S b}{I_{zz}} C_{n\beta} \right }$

E. GP Control

The original Generalized Pilot (GP) control was used in the development of the real-time aerodynamic modeling algorithm. The control was classically based and directly drove the primary control surfaces to track a wings level glide path at a constant pitch angle. Of interest, this control could operate concurrently with the real-time aerodynamic modeling algorithm without compromising the identification of the vehicle's stability and control derivatives in this wings-level flight condition. This was in contrast to the original modeling results with the adaptive nonlinear dynamic inversion control in maneuvering flight. One explanation was that the simpler control minimized its use of aero-dependent parameters (e.g., angle of attack) and instead regulated aircraft attitude error, much as an RC pilot would do, with minimal levels of angular feedback. As a result, vehicle inputs used to determine open-loop dynamics would be less correlated with aero-dependent parameters. The simpler control also did not use any modeling information.

In this section, modifications to the original GP control are described that were intended to determine if a simple control architecture defined in part by modeling results could avoid the perceived coupling problems of the more advanced adaptive nonlinear dynamic inversion control. These modifications include driving the control surfaces through the model-dependent control allocator, enabling maneuvering flight with excitation for identification, and defining a limited model-based augmentation for setting the time constant of the roll mode and the damping of the pitch and yaw axes in the closed loop.

Figure 1 displays a high-level diagram of the GP control. The outer-loop control, identical to that of the adaptive nonlinear dynamic inversion control law, converts commands in track angle and flight path angle into roll and pitch angle commands. An incremental pitch ladder command is added to the outer-loop pitch command for the purpose of indirectly expanding the range of angle of attack experienced to more fully cover the flight envelope for global aerodynamic modeling. The classical inner-loop control is a regulator with stability augmentation and produces commands for the vehicle's primary pseudo control surfaces (ailerons, elevator, and rudder).

$$\begin{aligned}\hat{\delta}_a &= K_{\phi_p} \phi_e + K_{\phi_i} \int \phi_e dt + K_p p, & \phi_e &\triangleq \text{Sat}_{\phi_{min}}^{\phi_{max}} (\phi_{cmd} - \phi) \\ \hat{\delta}_{ev} &= K_{\theta_p} \theta_e + K_{\theta_i} \int \theta_e dt + K_q q, & \theta_e &\triangleq \text{Sat}_{\theta_{min}}^{\theta_{max}} (\theta_{cmd} - \theta) \\ \hat{\delta}_{ru} &= K_{\beta_p} \beta_e + K_{\beta_i} \int \beta_e dt + K_r r, & \beta_e &\triangleq \text{Sat}_{\beta_{min}}^{\beta_{max}} (\beta_{cmd} - \beta)\end{aligned}$$

All error signals are limited between their specified minimum and maximum values by the saturation function; the proportional and integral gains for the errors are constant, fixed according to rules in [23]. In the transition from RC pilot to GP controls, the integrals are reset at the current surface position. The gains K_p , K_q , and K_r are modified using the real-time aerodynamic modeling results to respectively achieve an upper limit on roll mode time constant and certain lower limits on short period damping and Dutch Roll damping.

To illustrate, consider modifying K_r smoothly so that the Dutch Roll damping is not less than ζ_{DR}^* . The two-degree of freedom Dutch roll characteristic equation can be expressed as

$$s^2 + (-Y_\beta/V - N'_r)s - N'_\beta + Y_\beta N'_r/V = 0$$

Neglecting the effects of Y_β/V , define $-N'_r = 2\zeta_{DR}\omega_{DR}$ and $-N'_\beta = \omega_{DR}^2$, where ζ_{DR} and ω_{DR} are respectively the Dutch Roll damping and natural frequency. The required partial derivatives are taken from the global models of C_l and C_n generated by modeling, or

$$\begin{aligned}N'_\beta &= \left(\frac{\bar{q}Sb}{I_{zz}} \right) C'_{n\beta} \\ N'_r &= \left(\frac{\bar{q}Sb}{I_{zz}} \right) \left(\frac{b}{2V} \right) C'_{nr},\end{aligned}$$

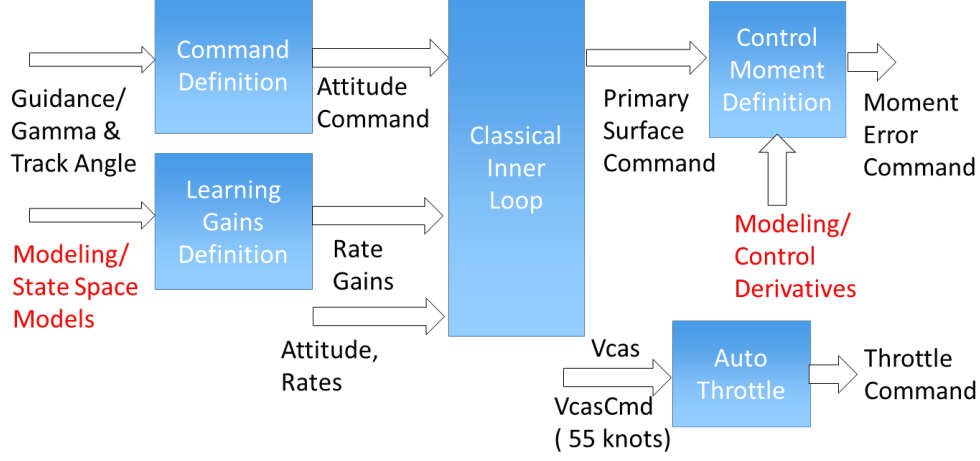


Fig. 1 GP Control Structure

where

$$C'_{n_\beta} = \left(1 - \frac{I_{xz}^2}{I_{xx}I_{zz}}\right)^{-1} \left(C_{n_\beta} + \frac{I_{xz}}{I_{xx}}C_{l_\beta}\right),$$

$$C'_{n_r} = \left(1 - \frac{I_{xz}^2}{I_{xx}I_{zz}}\right)^{-1} \left(C_{n_p} + \frac{I_{xz}}{I_{xx}}C_{l_p}\right),$$

using the standard nomenclature of [24]. Noting that the augmented damping satisfies $\zeta_{DR} = -(N'_r + k_r N'_{\delta_r})/(2\omega_{DR})$, where $\omega_{DR} = \sqrt{|N'_\beta|}$, the gain can be sequentially modified as follows: if at the i th iteration, $|\zeta_{DR} - \zeta_{DR}^*| > 0.02$ and $N'_{\delta_r} < 0$, define the target gain as $K_{r,target} = -(2\zeta_{DR}^* \omega_{DR} + N'_r)/N'_{\delta_r}$ and adjust the gain as

$$K_r(i+1) = k_r(i) + \lambda(K_{r,target} - K_r(i)),$$

where a small $\lambda > 0$ controls the step size. This expression will smoothly drive the gain in the correct direction to achieve the desired Dutch Roll damping. The resulting K_r is also limited so $K_r \in [0, K_{r,max}]$. An analogous procedure is used to modify K_p and K_q to achieve their respective goals.

Returning to Fig. 1, the pseudo primary surface commands are converted into moment commands to drive a non-incremental form of the control allocator. The non-incremental control allocator (see Section IV.F) approximately determines the surface deflections (not the change in surface deflection) to satisfy the moment command. The moment command is defined using partial derivatives of the global models of C_l , C_m , and C_n , taken with respect to the primary controls and evaluated at the current flight condition but with zero surface deflections:

$$M_{\delta,cmd} = \begin{bmatrix} \bar{q}bS \left(-C_{l_{\delta_{1a}}} \hat{\delta}_a + -C_{l_{\delta_{ra}}} \hat{\delta}_a + C_{l_{\delta_{ev}}} \hat{\delta}_{ev} + C_{l_{\delta_r}} \hat{\delta}_{ru}\right) \\ \bar{q}\bar{c}S \left(-C_{m_{\delta_{1a}}} \hat{\delta}_a + -C_{m_{\delta_{ra}}} \hat{\delta}_a + C_{m_{\delta_{ev}}} \hat{\delta}_{ev} + C_{m_{\delta_r}} \hat{\delta}_{ru}\right) \\ \bar{q}bS \left(-C_{n_{\delta_{1a}}} \hat{\delta}_a + -C_{n_{\delta_{ra}}} \hat{\delta}_a + C_{n_{\delta_{ev}}} \hat{\delta}_{ev} + C_{n_{\delta_r}} \hat{\delta}_{ru}\right) \end{bmatrix}.$$

It should be noted that the weighted least-squares solution of the non-incremental control allocator uses these same control derivatives, evaluated at zero surface deflections. This feature provides some robustness assuming the sign is correct on the primary pseudo control command: errors in sign of the key derivatives will be canceled in an allocator defined by the same derivative and will at least drive the surface in the correct direction.

Simulation results indicated that the GP controller architecture did not display the adverse model-control problem of the adaptive control. The PTI signals were less distorted compared to those of the adaptive control, and the estimated derivatives were closer to the ones in the actual database for wings-level flight. However, in maneuvering, the control

derivatives were still showing errors of 30%. Modifications to the regressor pools, discussed previously, fixed this problem. Although modeling with the GP controller produced better models, the update to the regressor pool also sufficiently improved the modeling results under the adaptive controller, to the point of eliminating the model-control coupling problem—the high-frequency control surface oscillation.

The GP controller is a viable L2F control method—although admittedly with more preset gains. Potentially, these regulator gains could be tuned with the incoming modeling information.

F. Control Allocation

Control allocation is the problem of determining aircraft control surface deflections to realize desired moments. Control allocation is generally required for over-actuated vehicles with a variety of redundant and multi-objective control effectors whose number exceeds the three moment components. The allocation of over-actuated vehicles has been formulated as a constrained optimization problem by many researchers [25–33]. Previously, many of these control allocation solutions were directed to specific reconfiguration problems, such as actuator failure states [26–29]. With the advent of more sophisticated health management systems envisioned for the NextGen aircraft [33], more constraints are being added to the allocation problem. These include limits on actuator travel, due to prognostics, and limits on monitored structural loading, due to detected damage. In response, real-time control allocation solutions possessing good convergence with acceptable computational requirements have been developed recently [25, 30–33]. Two of these solutions served as candidate allocation strategies for the L2F control.

To provide a brief overview, it is first noted that both approaches address the same allocation problem, but with different metrics. To be specific, let B denote the $3 \times n_\delta$ matrix of dimensional control derivatives defining the partial derivatives of moment with respect to the n_δ control surface deflections and δ_p denote some preferred control surface deflection vector (e.g. the null vector). Given some commanded moment $M_{\delta,cmd}$, the allocation problem is stated as follows. Find the deflection vector δ such that

$$J = \|B\delta - M_{\delta,cmd}\| + \epsilon\|\delta - \delta_p\|$$

is minimized, subject to $\delta_{min} \leq \delta \leq \delta_{max}$, where ϵ is small. In L2F, modeling determines B whereas the online control specifies $M_{\delta,cmd}$.

The control allocation problem defined here is a mixed optimization problem that combines the error and control minimization problems into a single problem with parameter ϵ . Minimizing error $B\delta - M_{\delta,cmd}$ is of the utmost importance, but for an over-actuated vehicle there are many solutions. Thus, the second term, to minimize control effort from some preferred position (e.g. to reduce drag), is added to the cost function to reduce the solution set. The metric used to define J is either the ℓ_1 or ℓ_2 norm. The ℓ_1 norm of a vector x is the sum of the absolute values of the elements of the vector,

$$\|x\|_1 = \sum_{i=1}^n |x_i|,$$

whereas the ℓ_2 norm is the usual Euclidean norm,

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}.$$

The cost function with either norm can be used to directly determine control surface position δ , or can be used in an incremental sense to find the change in control surface deflection. Specifically, find $\Delta\delta$ from its previous position δ^- so that

$$J = \|B\Delta\delta - \Delta M_{\delta,cmd}\| + \epsilon\|\Delta\delta - \Delta\delta_p\|$$

is minimized, subject to $\delta_{min} \leq \delta^- + \Delta\delta \leq \delta_{max}$. Here, $\Delta M_{\delta,cmd}$ is the change in moment required and the preferred $\Delta\delta_p$ could be the direction back to the null position. In the incremental implementation, B is evaluated locally at current surface deflection location and the commanded surface deflection vector is $\delta = \delta^- + \Delta\delta$. In the non-incremental implementation, B is evaluated at current conditions, but with zero surface deflections.

As stated, the mixed optimization problem combines the error and control minimization problems into a single problem through the use of a small parameter $\epsilon > 0$. For small ϵ , priority is given to error minimization over control

minimization, as is normally desired. Often, the combined problem may be solved faster, and with better numerical properties, than when the error and control minimization problems are solved sequentially [25]. It is possible to include a vector of weights inside the norms defining J , thereby allowing certain axes or control surfaces to be weighted differently in the cost function.

Computational resources available on modern aircraft make the use of optimal control allocation algorithms feasible in real time. An efficient algorithm to solve the mixed optimization problem with the ℓ_1 norm on the criterion was formulated by Bodson using linear programming approaches, providing guaranteed convergence to a solution in an acceptable period of time [25]. Timing data showed that solutions of the problem could comfortably be performed in real time, even for large numbers of actuators, and that the optimal solution improved performance significantly over simpler, approximate methods. The algorithm was based on the revised simplex method [34] with additional refinements, such as anti-cycling, as described in detail in [25].

Harkegård proposed an elegant solution of the optimal control allocation problem using the ℓ_2 norm and the theory of active sets [32]. The algorithm is very similar to the simplex algorithm used for ℓ_1 optimization, and it has the same advantage of completing in finite time and with a small number of iterations. For this project, Bodson's algorithm was used for the ℓ_1 control allocation solution and Harkegård's algorithm was used for the ℓ_2 solution.

During development, a sensitivity problem was discovered with the ℓ_1 control allocation. In the event that the estimated effectiveness of two surfaces is similar, and small fluctuations change which is most effective, rapid switching can occur. To demonstrate, consider an example with two surfaces of equal strength with estimated control derivatives $B = [1 + \nu \quad 1 - \nu]$, where ν is a small perturbation in the estimate. For $\nu > 0$, only the first surface will be used since it is more effective. Similarly, for $\nu < 0$, only the second surface will be used. For arbitrarily small perturbations in the estimate where ν changes signs, can cause deflection changes on the order of $M_{\delta,cmd}$ can occur. A situation similar to this example can often occur with symmetric surfaces, like ailerons. One approach to this problem might be to transform the deflections from, say, left and right to symmetric and differential. This, however, does not solve the problem, it simply moves it to another point in the space. When considering traditional aircraft surfaces, such as ailerons, this transformation should be sufficient as it is unlikely that the effectiveness estimates for symmetric and differential deflections will be similar. It becomes more challenging to effectively mitigate this risk with an appropriate transformation without sufficient a priori knowledge, which might be the case if one begins to consider novel aircraft configurations (one of the motivations for L2F). Since the ℓ_2 control allocation scheme does not have this sensitivity, it was selected for all of the flights described here.

V. Flight Test Results

A. Overview

Flight tests were performed over approximately two and a half weeks at Fort A.P. Hill in Virginia. Two vehicle types were used during the testing, a novel configuration glider, called Woodstock, and an RC kit pylon racer, called E1. The vehicles are depicted in Fig. 2. A total of 13 test flights were performed between the vehicles. A summary of the flights is given in Table 2.

Because they were unpowered, the Woodstock gliders required a launch mechanism. Due to cost and operational constraints, the vehicles were released from a tethered balloon [35]. Figure 3 shows the vehicle attached to the balloon,



Fig. 2 Two Woodstock aircraft (left) and the E1 aircraft (right)

Table 2 Descriptions of Test Flights

E1 Flight 2	Stand-alone NDI controller with separate model and control phases
E1 Flight 3	Adaptive NDI controller with separate model and control phases
E1 Flight 4	GP controller with separate model and control phases
E1 Flight 5	Adaptive NDI controller with short modeling-only phase before concurrent modeling and control
E1 Flight 6	Adaptive NDI controller with varying lengths of modeling-only phase before concurrent modeling and control
E1 Flight 9	Adaptive NDI controller with artificial destabilization in pitch ($\approx -10\%$ static margin)
E1 Flight 10	Adaptive NDI controller with artificial destabilization in roll (\approx neutrally stable)
E1 Flight 11	Adaptive NDI controller with artificial destabilization in pitch ($\approx -20\%$ static margin)
Woodstock Flight 1	Adaptive NDI controller
Woodstock Flight 2	Adaptive NDI controller

awaiting release. In the image, the tether has been highlighted with yellow and the wind direction can be seen. In order to prevent the vehicle from flying into the tether, a sail is attached to the balloon to orient it in the wind. The glider is then suspended such that the nose points away from the tether. One of the challenges imposed by this launch system is that the vehicle initially starts with an extremely large angle of attack, well beyond 90° , and very low dynamic pressure. This is outside the range where aerodynamics are generally well understood, making the probability of a good initial model very low, based on that data. Maintaining control of the vehicle immediately after launch is thus challenging.

From a test design perspective, E1 was more suitable for research purposes. Research tasks could be performed with little risk to the vehicle by toggling between the research control law and a stick-to-surface mode. In this way, an RC pilot could take off and land the vehicle without the additional challenges imposed by the balloon launch system. Also, by being able to convert to a known flyable configuration with the throw of a switch, the E1 aircraft was able to use some of its redundant surfaces to simulate a less stable vehicle. This allowed for testing of both stable and unstable systems, all in one platform.

B. NDI With and Without Adaptive Disturbance Rejection

Flight 2 began with an RC takeoff. Upon reaching an appropriate altitude, the learning mode was engaged with PTIs. Three laps were flown, two near the target speed and one at a slightly slower speed. At this point, the PTIs were disengaged and pitch, yaw, and roll doublets were performed by the pilot. Next, the autopilot containing the nonlinear dynamic inversion without the additional disturbance rejection loop was engaged; this is where the traces begin in the plots below. No significant transients were noticed, and the vehicle navigated to the waypoints. In Fig. 4, the system states are plotted in blue, with commanded values in red. The roll and pitch channels show good tracking performance, but a small bias can be seen in the yaw channel, particularly in the sideslip angle β . The autopilot mode was engaged several more times at different headings, bank angles, and speeds, all with similar results. The vehicle was returned to RC mode for landing.

After an RC takeoff, learning mode was engaged with PTIs for Flight 3. A lap was flown at medium, high, and low speeds with the PTIs active. Pitch, yaw, and roll doublets were performed after disabling the PTIs. The autopilot was then engaged with the nonlinear dynamic inversion and the additional disturbance rejection loop. No significant transients were observed as the vehicle began navigating waypoints as expected. Overall the tracking is good, as evidenced by Figures 5, but a small offset can be seen in the pitch channel, see θ and q . The autopilot was engaged in numerous conditions - roll angle, heading angle, high alt and slow speed, low alt and slow speed - all with similar results. PTIs were engaged during piloted approach to stalls. The autopilot was then engaged on an approach to stall, acting and navigating as expected. This is shown in Fig. 6 The vehicle was switched to RC mode for approach and



Fig. 3 Balloon launch system

landing.

Comparing Flights 2 and 3 using the average tracking error, Flight 3 shows improved lateral-directional tracking but decreased longitudinal tracking. The average tracking error for variable x is defined as

$$x_{e,avg}(k) = \frac{1}{k} \sum_{i=1}^k |x_{cmd}(i) - x(i)|.$$

In terms of the moment error commands, Flight 3 saw increased control usage in the pitch channel and decreased control usage in the roll and yaw channels. However, no real conclusions regarding the different control methods can be drawn. To truly separate the performance of the two controllers, the same model should be available to each. During these flights, the real-time modeling algorithm was identifying a model based on the data available, which varied based on the flight. Moreover, the control system being used impacts the modeling results (cf. [2]), meaning that this feedback loop needs to be broken to investigate the effects of the controllers.

Another obfuscating factor was the lack of quality real-time thrust modeling and control on these flights. Without the appropriate regressors, the aerodynamic terms need to absorb the thrust effects. Looking at the dynamics for angle of attack, this time including thrust terms,

$$\dot{\alpha} = q - \tan(\beta) (\cos(\alpha)p + \sin(\alpha)r) + \frac{1}{mV \cos(\beta)} (-L + mg \cos(\gamma) \cos(\mu) - T_x \sin(\alpha) + T_z \cos(\alpha)),$$

it is clear that thrust can play a major role in the longitudinal dynamics, especially since the motor on E1 was strong relative to the vehicle's mass. This portion of controller was designed for a glider, not accounting for thrust, so these effects will naturally show up. Additionally, the thrust controller that was implemented resulted in sloppy speed control with large oscillations in the thrust commands. Overall, separate experiments should be carefully designed if a true comparison of the control methods is desired.

C. GP in Flight

In the GP development, it was assumed that the modeling would always run concurrently with the control system. There seemed to be no need to duplicate calculations in the control module that were performed in modeling module. A particularly important one produces the dimensional dynamic derivatives, i.e. L'_p , L'_{δ_a} , N'_p , N'_β , N'_{δ_r} , M_q , M_{δ_e} , and Z_α , used to determine angular rate feedback gain—each of which is a function of dynamic pressure. Unfortunately, when modeling is disabled, the outputs are fixed at their last values. This is not a problem when the coefficients and regressor

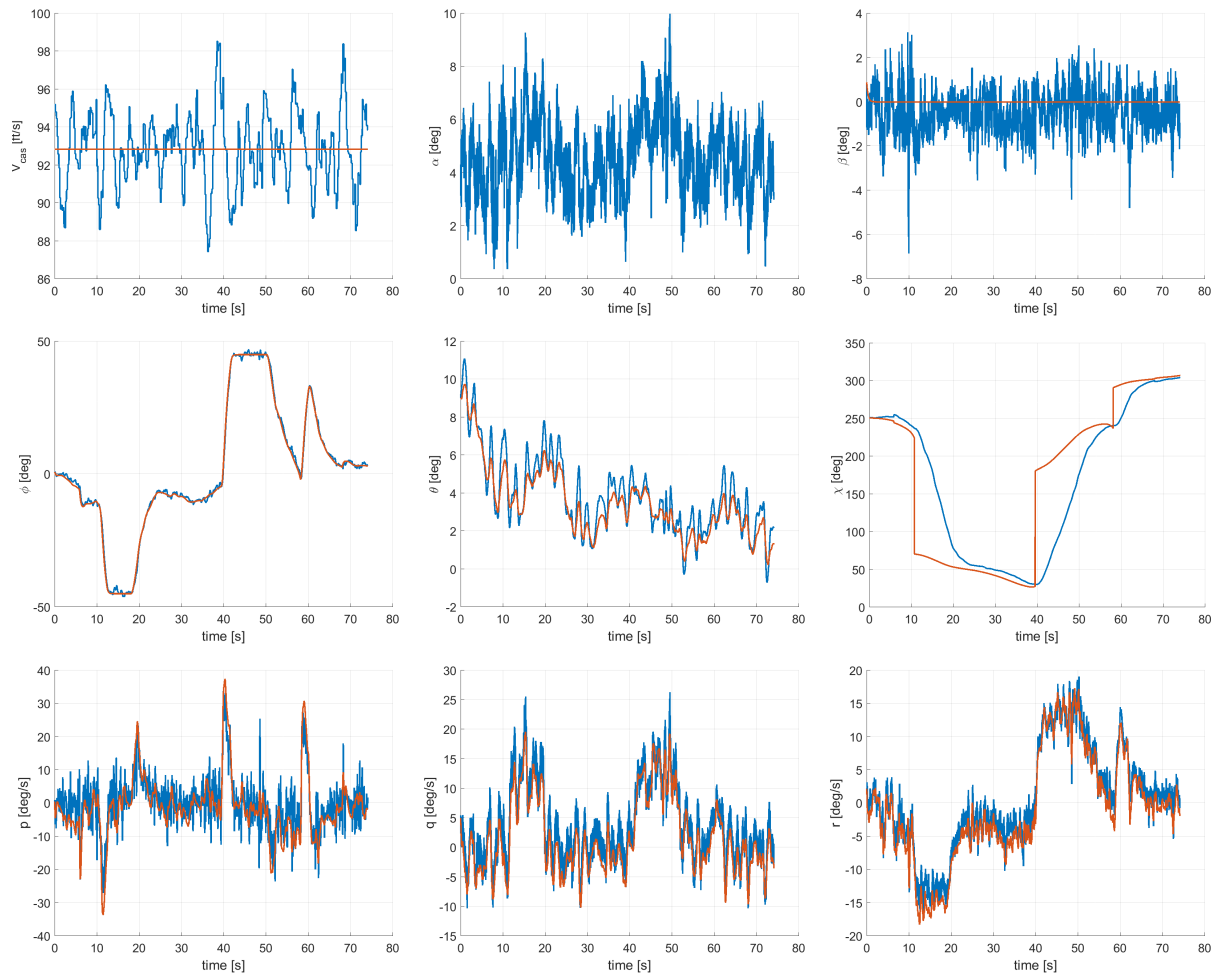


Fig. 4 Flight 2 Non-adaptive NDI Controller - Nominal Trajectory

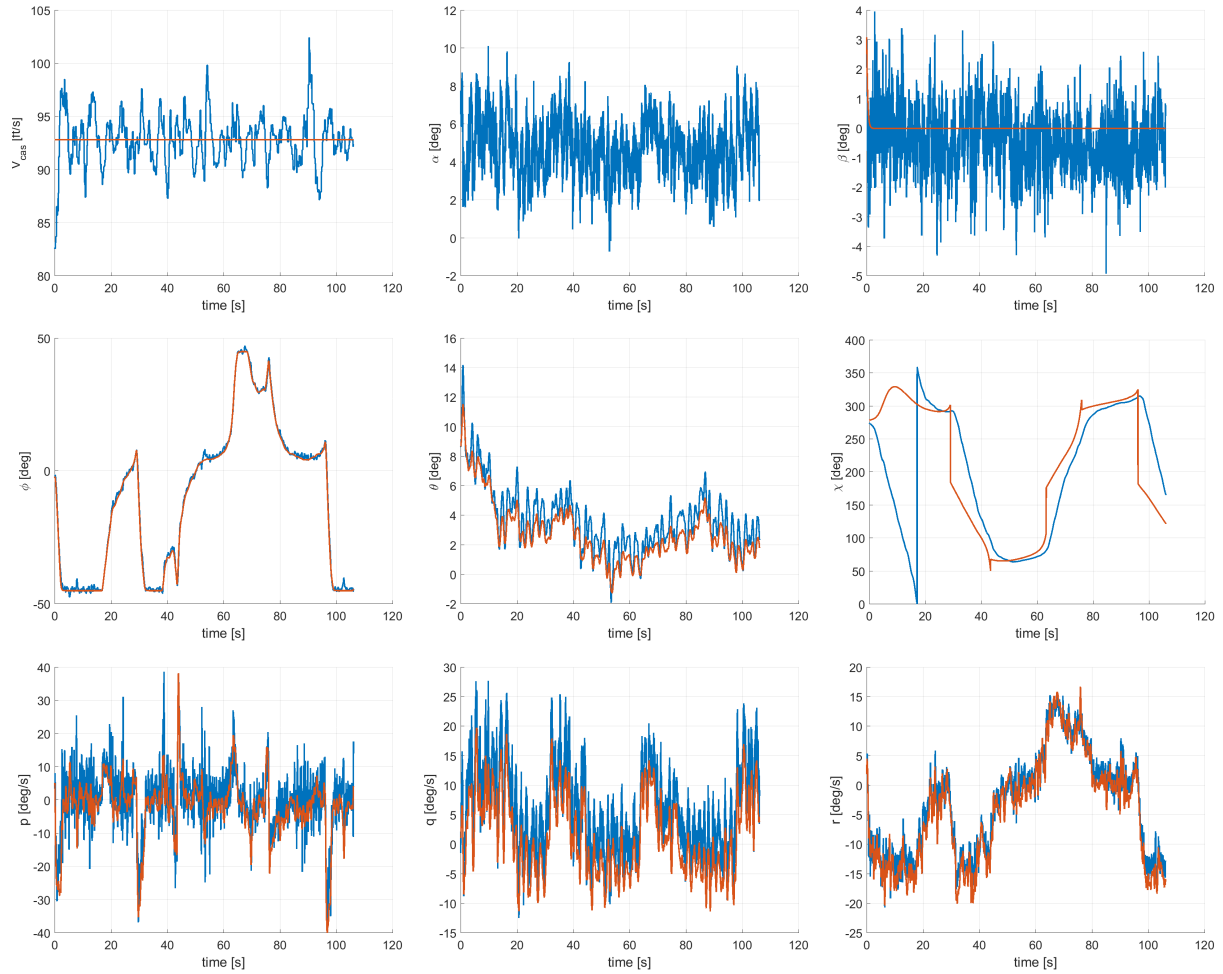


Fig. 5 Flight 3 Adaptive NDI Controller - Nominal Trajectory

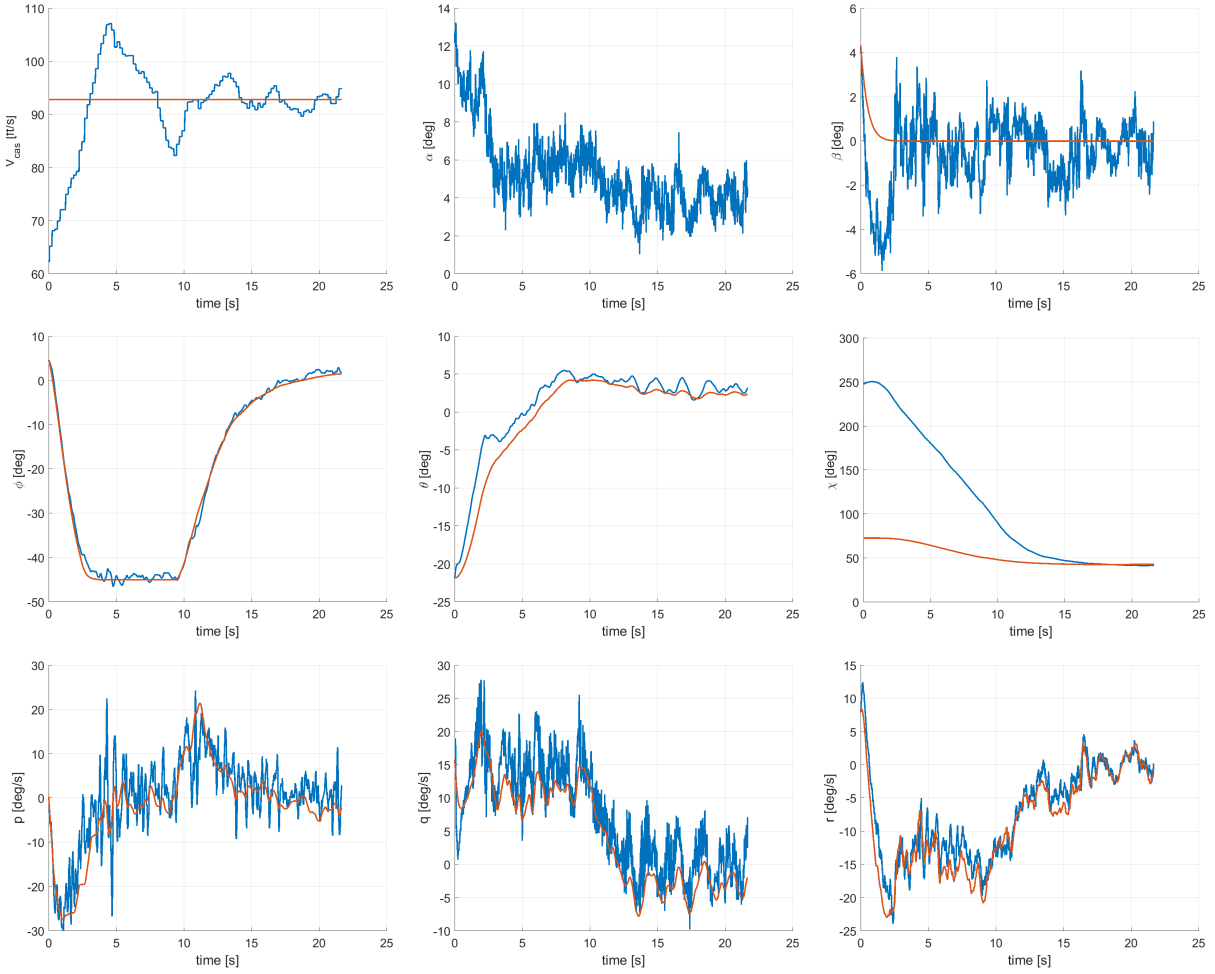


Fig. 6 Flight 3 Adaptive NDI Controller - Approach to Stall

definitions are fixed for the global model output. Current values of both dimensional and non-dimensional derivatives can be easily obtained from the fixed global functions using current explanatory variables taken from the IMU bus. This was not done in the GP controller.

Prior to deployment, it was determined, due to the potential that the model-control coupling problem might still occur, that for the first few flights, modeling would be done first with the RC pilot in control and then held while the control used the fixed modeling information. The limited testing of GP did not take into account a hand-off condition vastly different than the target condition, defined by the constant 55 knots tracked by the E1 vehicle's auto-throttle. Because of this, the dimensional dynamic derivatives that were used were not always representative for the current flight condition. Two sets of plots are provided in Fig. 7 and Fig. 8 to illustrate what happened. In Fig. 7, the release occurred at the target speed, the vehicle tracked the guidance command with the gains fixed at $K_p = 0.06$, $K_q = 0.01$, and $K_r = 0.09$. Later in the flight when the release occurred at half the target speed, shown in Fig. 8, the held dimensional derivatives produced gains that were too large when the auto-throttle drove the vehicle to the constant target speed. The vehicle went into an unstable roll oscillation, and the RC pilot took the command back. The destabilizing gains were $K_p = 0.5$, $K_q = 0.18$, and $K_r = 0.18$.

During this deployment, the GP control flew once. The obvious lesson is that dimensional derivatives of the global model, used to define the control law, need to be constructed outside of the modeling module using the current explanatory variables from the IMU. The held non-dimensional derivatives used to convert the pseudo primary controls to moment commands likely were not a problem because they were dimensionalized in the control module.

D. Concurrent vs Non-concurrent

Although the approximation of the total moments plays an important role in the adaptive nonlinear dynamic inversion, stability and control derivatives taken from these moment approximations play an equally important part in control for setting the desired dynamics, adjusting gains, and defining the control allocator. The least-squares modeling approach that was taken builds on the accumulated information gathered to produce global models using model terms deemed significant for approximating the dependence of total force and moment non-dimensional coefficients on the explanatory variables. Moreover, the best model will be the one based on all the data at the end of the learning/modeling period. It is also of interest to the control designer to get some idea of the spread expected in the terms that determine key derivatives. In this section, terms from the C_l models obtained over four flights (E1 Flights 3-6) are considered. These flights are of particular interest because they demonstrate the modeling results under pilot control, as well as with the adaptive nonlinear dynamic inversion controller running concurrently.

The C_l regressor pool contained 25 terms, which included a bias term. Figure 9 shows the history of the learned coefficients corresponding to some of the linear terms of the model over the four flights. The first two flights, in red and blue, each have 10 periods of learning, where each period of the flight builds on what was learned in the previous periods. In these two flights, modeling was not operating concurrently with the adaptive nonlinear dynamic inversion control; the RC pilot was in control of the vehicle. At the beginning of each flight, modeling began from its initial state. In the first and ninth period of learning, designated with triangles, PTIs were exciting the surfaces. In the first period, the vehicle was flying at roughly 55 knots, whereas the ninth period examined slower speeds at higher angles of attack. During other learning periods, designated by circles, modeling was solely based on whatever control inputs the RC pilot made. Results from the third flight are designated by a black square. This flight occurred on the same day as the first two and consisted of an extended period of learning equal to the sum of the shorter learning periods of either of the first two flights. In contrast, modeling ran for the first time concurrently with the adaptive nonlinear dynamic inversion control and PTIs running throughout the learning period. Results from the fourth flight, shown as magenta squares, involved resetting the data memory to zero and setting selected linear model parameters to initial values (guesses) prior to each of the learning periods, which were roughly equal to the first learning period of the first two flights. In this fourth flight, modeling ran concurrently with the adaptive nonlinear dynamic control.

As one would expect, model parameters from the first two flights that built upon the learning of the previous period had few discontinuities. For the most part, progression of terms for the left and right control surfaces mirrored one another. It is interesting that the left and right aileron appear to be losing effectiveness as learning progresses in the first flight, whereas in the other flights shown, the aileron effectiveness from the linear terms is roughly in the same neighborhood—this includes concurrent operations of model and control. The effectiveness of the flaps appears roughly the same over the first three flights, with some discontinuous increase for the later learning periods of the second flight. Higher effectiveness is modeled for the concurrent operations with shorter learning periods of the fourth flight. The consequence of identifying a control derivative at a much lower value than the actual effectiveness is one of the

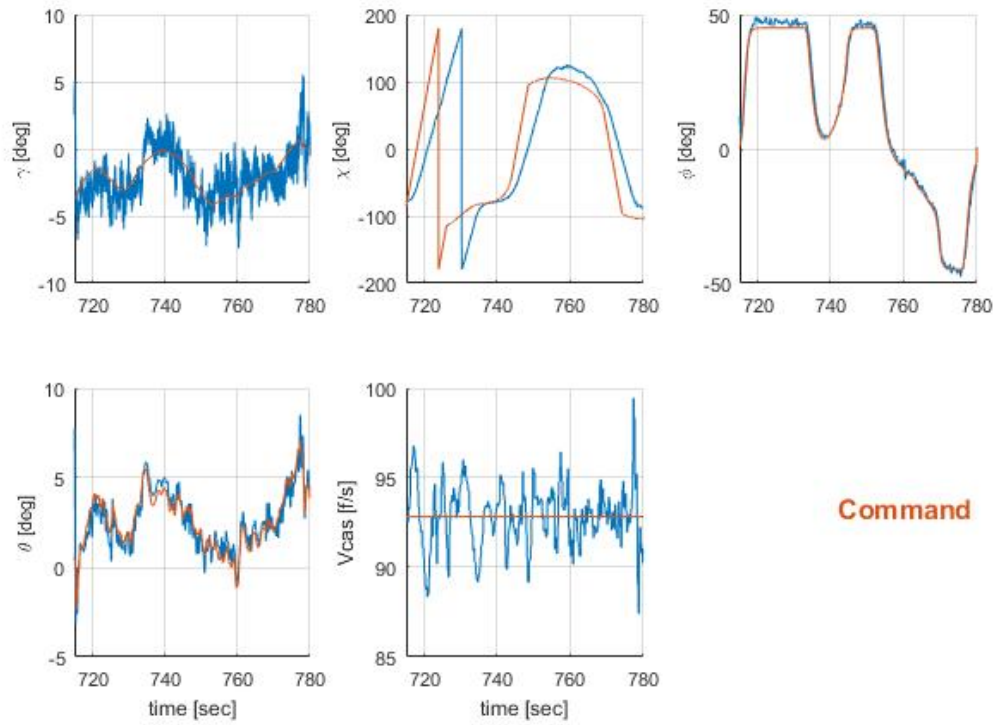


Fig. 7 GP with modeling held at target speed

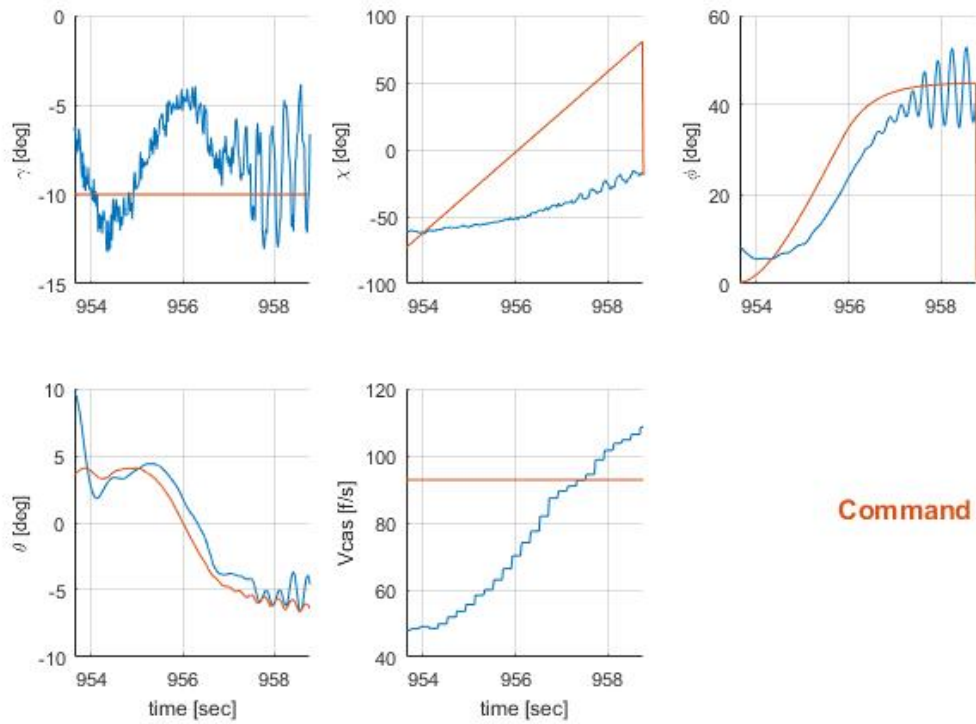


Fig. 8 GP with modeling held at half target speed

allocation overdriving the control surface in meeting a commanded moment. Because no such behavior was observed in flight (for the last two flights shown; the RC pilot directly controlled the surfaces in the first two), it can be assumed that these values well approximate their portion of the control derivative.

With regard to the key metric for stability in the roll axis, the roll damping C_{l_p} , the trend was similar to that for the ailerons. The first flight shows a migration of 30% reduction in the estimated roll damping—effectively a 30% reduction in the magnitude of the roll-mode pole, or a roll-mode time constant that is 3.33 times longer. Again, in the results for the other flights shown, including the ones with concurrent model/control operations, this parameter does not display this trend but is clustered in roughly the same region. Consequently it does not appear that controls has much of an impact on the identification of this key stability derivative.

It should be mentioned that other regressors in the model contribute to the stability and control derivatives when evaluated on the vehicle trajectory. A complete list of the C_l candidate regressor pool is provided in Table 3. However, for control purposes, the linear design point for defining the desired dynamics and gains is often taken to be with the angular rates, control deflections, and sideslip angle set to zero. However, for control purposes, the angular rates, control deflections, and sideslip angle, in particular, are often set to zero in the derivative calculations, in order to provide smooth derivatives for defining the desired dynamics and gains. As a result, the key modeling parameters for the controller are the linear ones, and from the flight data present, concurrent operations of modeling and the nonlinear adaptive control did not appear to be a problem.

E. Unstable Vehicle

Whereas earlier flights were successful with a stable vehicle, additional flights were designed to test the system with an unstable vehicle. Flight 9 was the first of two flights that saw a destabilization of the pitch axis. Using a hidden feedback loop, the angle-of-attack was fed to the left elevator to produce roughly -10% static margin. The nominal vehicle was taken up to altitude under control of the RC pilot. The learning mode was engaged, activating destabilizing feedback, the PTIs, and the autopilot. The vehicle stabilized and navigated to the waypoint. This was repeated with similar results. The RC pilot also flew the destabilized aircraft. He was able to compensate with large stick inputs during straight and level flight and moderate angle-of-attack and pitch-attitude excursions. The nominal configuration was resumed for approach and landing.

Flight 11 was similar to Flight 9 but with a larger instability in the pitch axis, about -16.4% static margin. The nominal vehicle was taken up to altitude before engaging the destabilizing feedback, PTIs, and autopilot. A couple pitch oscillations occurred, but the vehicle recovered and flew to the waypoint. This was repeated with similar results. When the RC pilot flew the destabilized aircraft, he was barely able to control it using full or almost-full stick deflections. The system was reset to its nominal configuration for approach and landing.

Plots of the autonomous system’s tracking performance from Flight 11 are shown in Fig. 10. After some large errors at the beginning of the flight, the flight parameters settled and tracked their respective commands. Figure 11 shows an overlay of the pilot’s flight data (red) and that of the autonomous system (blue). Despite the large inputs from the pilot, large pitch oscillations still occur. The real-time modeling, however, quickly determines that the vehicle is unstable (positive C_{M_α} estimate), and the adaptive control modifies the control gain. Note that there is inherently a delay in the modeling results because time must be allowed to collect sufficient informative data. Essentially, the modeling algorithm must see the behavior in the data before that behavior can be appropriately modeled. For the first few seconds, the controller is still operating based on a stable aircraft model (the initial guess). This is where the robustness of the controller is critical. At the beginning, the controller struggles to achieve good tracking performance, as evidenced by the initially large pitch excursion. However, as the system has time to learn, the controller updates, performance improves, and smooth, level flight is achieved.

In Flight 10, the roll axis of the vehicle was destabilized by feeding the roll rate back to the inboard flaps. Although the initial target was a time to double of 0.5 seconds, the initial feedback gain used only made the aircraft roughly

Table 3 C_l regressor pool for E1

CI Regressors									
Linear	bias	β	δ_{ru}	\hat{r}	δ_{lf}	δ_{rf}	δ_{la}	δ_{ra}	δ_{ev}
Higher	$\beta\alpha$	$\beta\hat{p}$	$\beta\hat{r}$	$\beta\delta_{lf}$	$\beta\delta_{rf}$	$\beta\delta_{la}$	$\beta\delta_{ra}$	$\beta\delta_{ru}$	
Order	$\hat{p}\hat{r}$	$\hat{p}\delta_{lf}$	$\hat{p}\delta_{rf}$	$\hat{p}\delta_{la}$	$\hat{p}\delta_{ra}$	$\hat{p}\delta_{ru}$	$\hat{r}\delta_{ru}$		

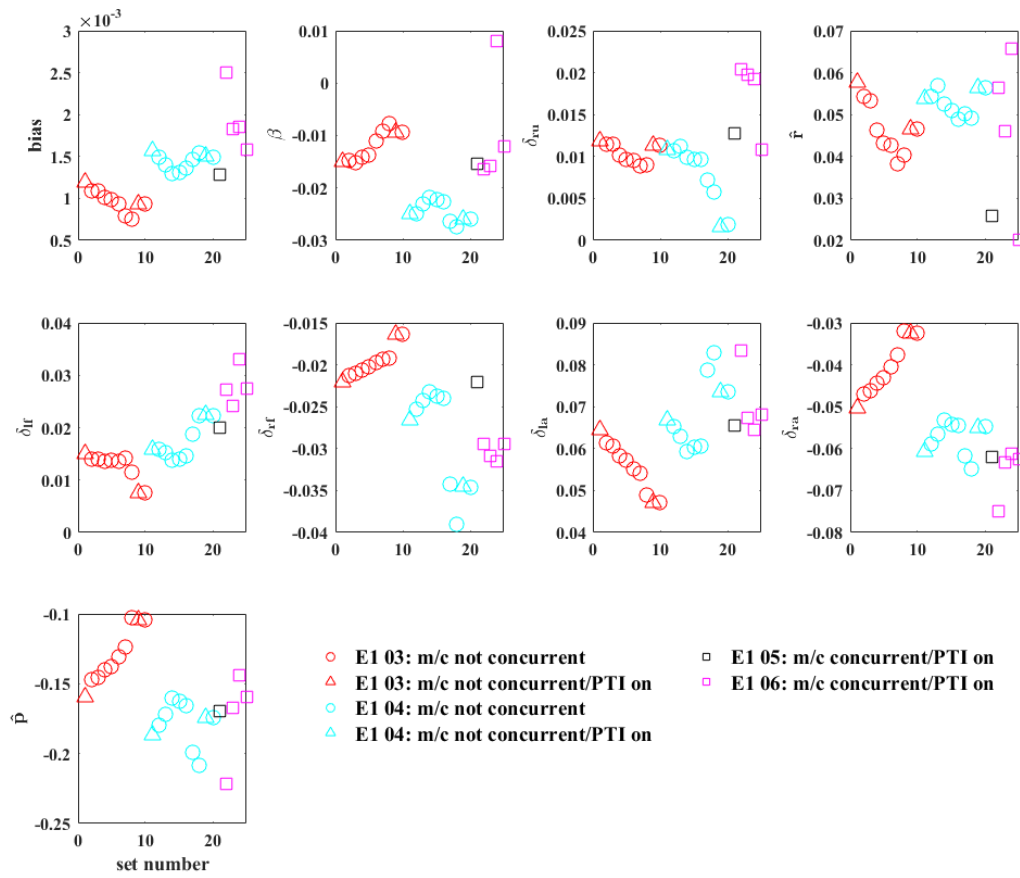


Fig. 9 Linear terms of C_l model identified in flight

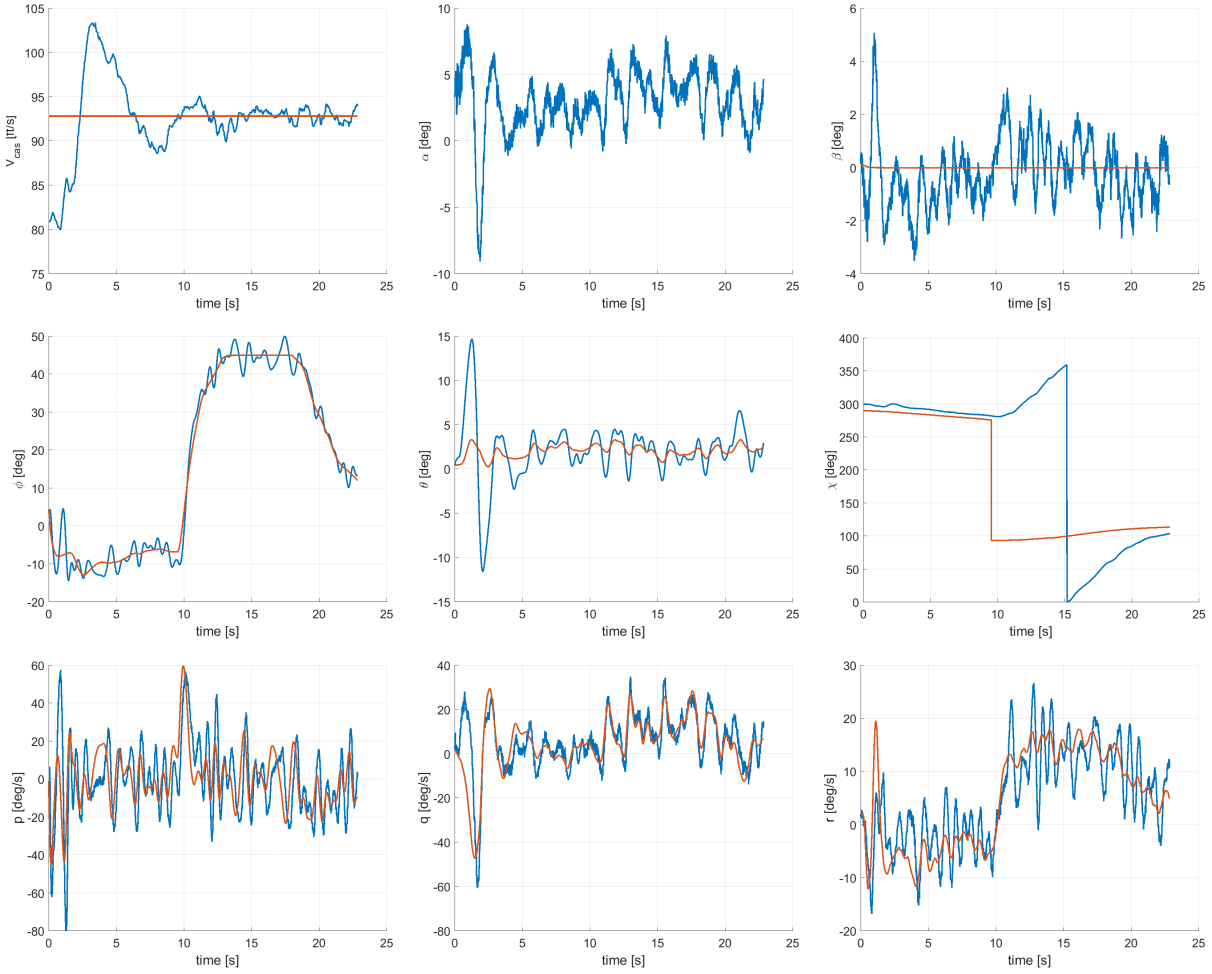


Fig. 10 Flight 11 Adaptive NDI Controller - Vehicle Pitch Axis Destabilization

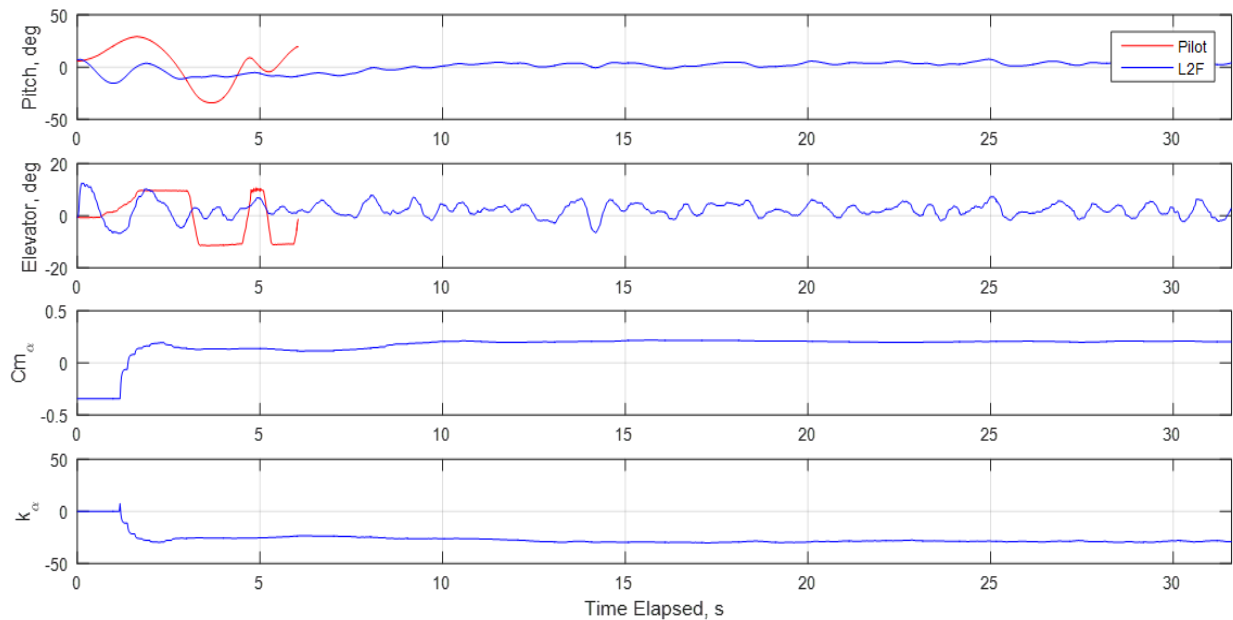


Fig. 11 Overlay of Flight 11 Pilot Data with Autonomous Data

neutrally stable, which was still a degradation of the vehicle's innate stability. After takeoff, the nominal vehicle was flown to altitude before engaging the destabilizing feedback, the PTIs, and the autopilot. During both attempts, the vehicle stabilized after a few roll oscillations and navigated as expected. When the RC pilot took control of the destabilized aircraft the first time, the vehicle was stabilized with some large roll oscillations. On the second attempt, the RC pilot maintained better control but still noted that it was difficult to fly. After reverting to the nominal configuration, the aircraft was brought in for a landing.

The tracking data for Flight 10 is shown Fig. 12. Once again, the large tracking errors at the beginning, before the Learn-to-Fly algorithm has had a chance to learn, quickly die down as more information becomes available and the adaptive control operates.

F. Woodstock Flights

The first Woodstock flight effectively ended upon release from the balloon. A nose-up motion was immediately commanded by the controller, causing the actuated tail to rise. This caused the tail to strike the support structure on the balloon, stripping the servo horn and rendering the tail inoperable. The vehicle tucked nose under and entered a stable inverted spiral to the ground.

The second Woodstock flight successfully cleared the carrier. Recovery from the awkward attitude began as the vehicle brought its nose up. However, due to limitations in the test range, a turn was soon commanded. It seems as if the vehicle had not yet determined how to control itself and dropped into a stalled spiral. Compounding this, a fault in the altimeter caused spurious readings that triggered the control law's flight path angle tracking mode. As explained above, this mode is controlled through pitch attitude. Because it was designed for the landing phase, this control mode inherently assumes that the vehicle is not stalled. Typically a stalled aircraft will require the nose to be brought down to reduce the angle of attack before it can bring the nose up to achieve a desired glide path. This was not the case during this stalled spiral. The control law provided a nose up command, trying to achieve the desired glide path, and instead kept the vehicle stalled. As the vehicle descended, the system was still attempting to establish control; at some point, the direction of the spiral reversed. However, the vehicle was ultimately unable to regain control of itself.

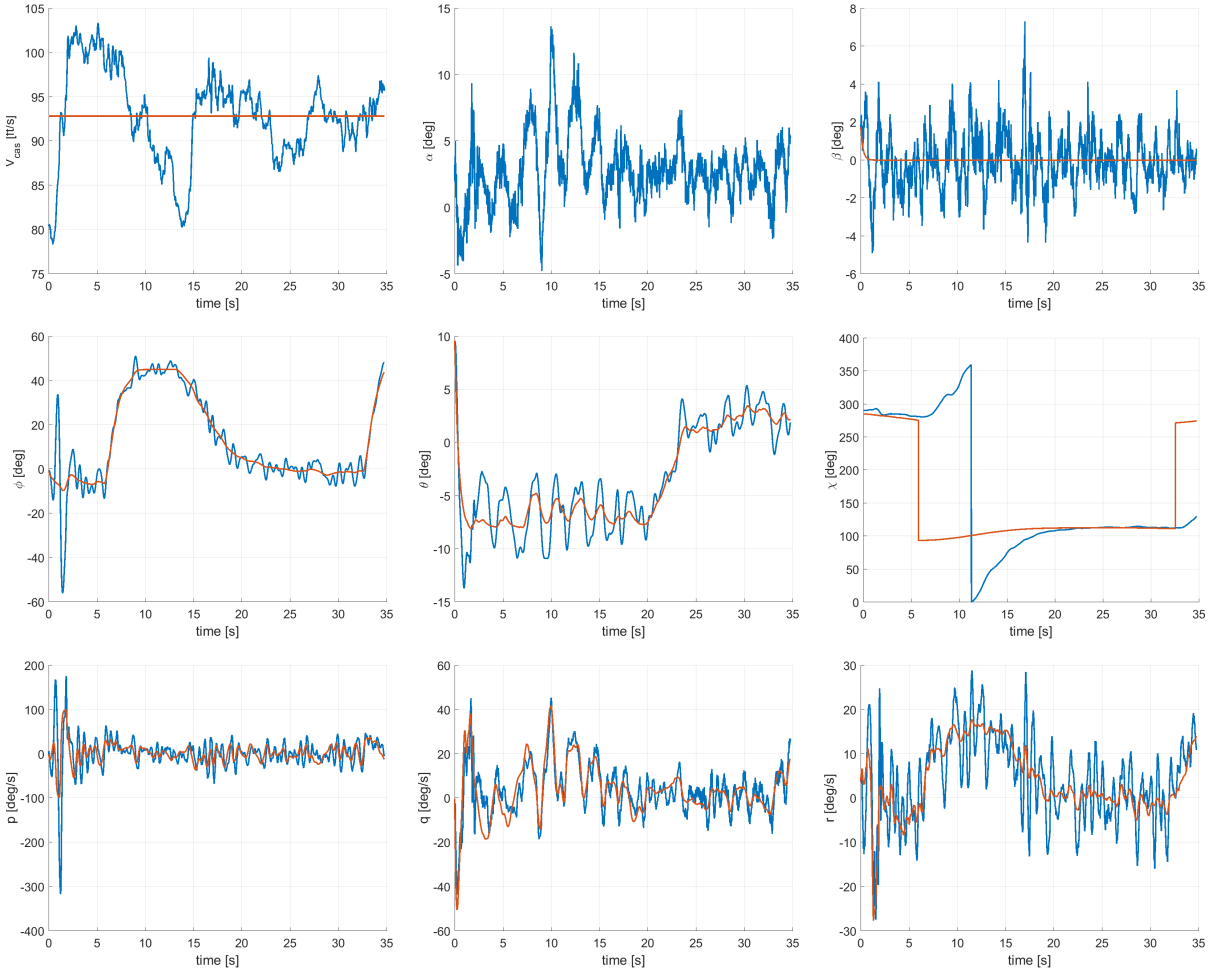


Fig. 12 Flight 10 Adaptive NDI Controller - Vehicle Roll Axis Destabilization

VI. Conclusions

The first conclusion that can be drawn from this work is that the Learn-to-Fly paradigm is feasible. The E1 flight tests have shown that the framework has been successfully applied to a stable or unstable aircraft. However, the Woodstock flights demonstrated that the current system cannot be universally applied with success and that further study is required. A few control architectures were tested, but no conclusions can be drawn regarding which is most appropriate for this mission. Additional tests should be performed, being careful to control the test conditions, including what information is available to the controllers. A question remains as to how robust the controller needs to be to the initial guess of the vehicle's dynamics. For the E1 flights presented here, the initial guess was based on a completely different aircraft, though of similar size. In practice, when testing new vehicle designs, it is expected that a better guess will be available, based on the design criteria and parameters.

While overall it seems that the real-time aerodynamic modeling and the control laws can run concurrently, their interaction is still not completely understood. Having an extensive set of candidate model terms is important, and excitation is needed to produce informative data for good modeling results, but, based on the data, this is still dependent on the controller. Specifically, appropriate model terms should be included for thrust so that the effects of thrust are not lumped together with other aerodynamic terms. It would be best if the vehicle did not need to maneuver while learning its dynamics, but this is dependent on the test range available. Ideally, there would be some executive function that could balance the modeling and control objectives, allowing sufficient learning before navigating.

Despite the successful demonstration of feasibility of the Learn-to-Fly paradigm, future research is required to develop and study the tools that enable it.

References

- [1] Heim, E. H., Viken, E. M., Brandon, J. M., and Croom, M. A., "NASA's Learn-to-Fly Project Overview," *AIAA Atmospheric Flight Mechanics Conference*, Atlanta, GA, 2018 (to be published).
- [2] Morelli, E. A., "Practical Aspects of Real-Time Modeling for the Learn-to-Fly Concept," *AIAA Atmospheric Flight Mechanics Conference*, Atlanta, GA, 2018 (to be published).
- [3] Foster, J. V., "Autonomous Guidance Algorithms for NASA Learn-to-Fly Technology Development," *AIAA Atmospheric Flight Mechanics Conference*, Atlanta, GA, 2018 (to be published).
- [4] Weinstein, R., Hubbard, J. E., and Cunningham, M. A., "Fuzzy Modeling and Parallel Distributed Compensation for Aircraft Flight Control from Simulated Flight Data," *AIAA Atmospheric Flight Mechanics Conference*, Atlanta, GA, 2018 (to be published).
- [5] Grauer, J. A., "A Learn-to-Fly Approach for Adaptively Tuning Flight Control Systems," *AIAA Atmospheric Flight Mechanics Conference*, Atlanta, GA, 2018 (to be published).
- [6] Ward, D., Monaco, J., Barron, R., Bird, R., Virnig, J., and Landers, T., "Self-Designing Controller: Design, Simulation, and Flight Test Evaluation," SBIR Phase II Final Technical Report Contract F49620-94-C-0087, Barron Associates, Inc., November 1996.
- [7] Chandler, P., Mears, M., and Pachter, M., "On-Line Optimizing Networks for Reconfigurable Control," *Proceedings of 32nd Conference on Decision and Control*, San Antonio, TX, 1993, pp. 2272–2277.
- [8] Chandler, P., Pachter, M., and Mears, M., "On-Line Optimizing Networks for Reconfigurable Control," *AIAA Journal of Guidance, Control, and Dynamics*, , No. 18, 1995, pp. 516–524.
- [9] Durham, W. C., and Bordignon, K. A., "Multiple Control Effector Rate Limiting," *AIAA Journal of Guidance, Control, and Dynamics*, , No. 19, 1996, pp. 30–37.
- [10] Bolling, J. G., and Durham, W. C., "Control Allocation with Adaptive Failure Control," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Scottsdale, AZ, 1997.
- [11] Buffington, J. M., "Tailless Aircraft Control Allocation," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, 1997.
- [12] Wise, K. A., Brinkner, J. S., Calise, A. J., Enns, D. F., R., M., and Voulgaris, "Direct Adaptive Reconfigurable Flight Control for a Tailless Advanced Fighter Aircraft," *International Journal of Robust and Nonlinear Control*, 1999, pp. 999–1012.

- [13] Kim, B. S., and Calise, A. J., "Nonlinear Flight Control Using Neural Networks," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Scottsdale, AZ, 1997.
- [14] Baker, W. L., and Farrell, J. A., "Learning Augmented Flight Control for High Performance Aircraft," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, 1991, pp. 347–358.
- [15] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, MA, 1998.
- [16] Eberhardt, R., and Ward, D., "Indirect Adaptive Flight Control System Interactions," *International Journal of Robust Nonlinear Control*, Vol. 9, 1999, pp. 1013–1031.
- [17] Klein, V., and Morelli, E. A., *Aircraft System Identification – Theory and Practice*, American Institute of Aeronautics and Astronautics, 2006.
- [18] Gentleman, W. M., "Regression Problems and the QR-decomposition," *Bulletin of the IMA*, 1975, pp. 195–197.
- [19] Gentleman, W. M., "Least Squares Computation by Givens Transformations Without Square Roots," *Institute of Mathematics and Its Applications*, , No. 12, 1973, pp. 329–336.
- [20] Montgomery, D. C., *Design and Analysis of Experiments*, 5th ed., John Wiley & Sons, Inc., 2002.
- [21] Snell, S. A., "Nonlinear Dynamic-Inversion Flight Control of Supermaneuverable Aircraft," Ph.D. thesis, University of Minnesota, October 1991.
- [22] Steven, B. L., and Lewis, F. L., *Aircraft Control and Simulation*, 2nd ed., John Wiley & Sons, Inc., 2003.
- [23] Beard, R. W., and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, 2012.
- [24] McRuer, D., Ashkenas, I., and Graham, D., *Aircraft Dynamics and Automatic Control*, Princeton University Press, 1973.
- [25] Bodson, M., "Evaluation of Optimization Methods for Control Allocation," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711.
- [26] Durham, W., "Constrained Control Allocation," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 717–725.
- [27] Enns, D., "Control Allocation Approaches," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Boston, MA, 1998.
- [28] Buffington, J., "Modular Control Law Design for the Innovative Control Effectors (ICE) Tailless Fighter Aircraft Configuration 101-3," Tech. Rep. AFRL-VA-WP-TR-1999-3057, Air Force Research Laboratory, Wright-Patterson AFB, OH, 1999.
- [29] Burken, J. J., Lu, P., Wu, Z., and Bahm, C., "Two Reconfigurable Flight-Control Design Methods: Robust Servomechanism and Control Allocation," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, 2001, pp. 482–493.
- [30] Petersen, J., and Bodson, M., "Interior-Point Algorithms for Control Allocation," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 471–480.
- [31] Petersen, J., and Bodson, M., "Constrained Quadratic Programming Techniques for Control Allocation," *IEEE Transactions on Control Systems Technology*, Vol. 14, No. 1, 2006, pp. 91–98.
- [32] Harkegard, O., "Efficient Active Set Algorithms for Solving Constrained Least Squares Problems in Aircraft Control Allocation," *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 2, Las Vegas, NV, 2002, pp. 1295–1300.
- [33] Frost, S. A., Bodson, M., Burken, J. J., Jutte, C. V., Taylor, B. R., and Trinh, K. V., "Flight Control with Optimal Control Allocation Incorporating Structural Load Feedback," *Journal of Aerospace Information Systems*, Vol. 12, No. 12, 2015, pp. 825–834.
- [34] Luenberger, D., *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1984.
- [35] Riddick, S. E., Busan, R. C., Cox, D. E., and Laughter, S. A., "Learn-to-Fly Test Setup and Concept of Operations," *AIAA Atmospheric Flight Mechanics Conference*, Atlanta, GA, 2018 (to be published).