JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 27, No. 6, November–December 2004

Integrated Adaptive Guidance and Control for Re-Entry Vehicles with Flight-Test Results

John D. Schierman,* David G. Ward,† Jason R. Hull,‡ and Neha Gandhi§
Barron Associates, Inc., Charlottesville, Virginia 22901
and
Michael W. Oppenheimer¶ and David B. Doman**
U.S. Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433-7531

To enable autonomous operation of future reusable launch vehicles, reconfiguration technologies will be needed to facilitate mission recovery following a major anomalous event. The Air Force's Integrated Adaptive Guidance and Control program developed such a system for Boeing's X-40A, and the total in-flight simulator research aircraft was employed to flight test the algorithms during approach and landing. The inner loop employs a model-following/dynamic-inversion approach with optimal control allocation to account for control-surface failures. Further, the reference-model bandwidth is reduced if the control authority in any one axis is depleted as a result of control effector saturation. A backstepping approach is utilized for the guidance law, with proportional feedback gains that adapt to changes in the reference model bandwidth. The trajectory-reshaping algorithm is known as the optimum-path-to-go methodology. Here, a trajectory database is precomputed off line to cover all variations under consideration. An efficient representation of this database is then interrogated in flight to rapidly find the "best" reshaped trajectory, based on the current state of the vehicle's control capabilities. The main goal of the flight-test program was to demonstrate the benefits of integrating trajectory reshaping with the essential elements of control reconfiguration and guidance adaptation. The results indicate that for more severe, multiple control failures, control reconfiguration, guidance adaptation, and trajectory reshaping are all needed to recover the mission.

Nomenclature

C_D, C_L	=	drag and lift coefficients
D	=	drag, lb
g	=	acceleration of gravity, ft/s^2
H	=	altitude, ft
I	=	mass moment of inertia matrix, slug $\cdot$ ft^2
L	=	lift, lb
L, M, N	=	rolling, pitching, yawing moments, ft $\cdot$ lb
M	=	Mach number
m	=	vehicle mass, slugs
p, q, r	=	rolling, pitching, yawing angular velocities, rad/s
$\bar{q}$	=	dynamic pressure, psf
S	=	wing area, ft^2
V	=	velocity, fps
X	=	downrange, n mile
α	=	angle of attack, deg
γ	=	flight path angle, deg
δ	=	control surface deflection vector, deg
ω	=	angular velocity vector, rad/s

Received 25 April 2004; revision received 8 July 2004; accepted for publication 8 July 2004.

*Senior Research Scientist, 1410 Sachem Place, Suite 202; schierman@bainet.com. Associate Fellow AIAA.
†Senior Research Scientist, 1410 Sachem Place, Suite 202; ward@bainet.com. Senior Member AIAA.
‡Research Scientist, 1410 Sachem Place, Suite 202; hull@bainet.com.
§Research Associate, 1410 Sachem Place, Suite 202; gandhi@bainet.com.
¶Electronics Engineer, 2210 Eighth Street, Building 146, Room 305; Michael.Oppenheimer@wpafb.af.mil. Member AIAA.
**Senior Aerospace Engineer, 2210 Eighth Street, Building 146, Room 305; David.Doman@wpafb.af.mil. Senior Member AIAA.

Introduction

IN recent years, NASA and the Department of Defense have focused significant resources on developing advanced launch technologies. Efforts continue to retire the space shuttle and develop new manned launch/orbital vehicles, as well as completely autonomous space transportation systems to service the International Space Station.[1,2] An overriding concern is to develop launch platforms that are significantly safer, more reliable, and cheaper to develop/build.[3] Reconfiguration capabilities for guidance and control systems play an integral part in achieving these goals, especially for autonomous systems that do not have the advantage of human reasoning and adaptation under anomalous circumstances (such as control-effector failures).

Due to weight, size, and cost restrictions, launch vehicles typically utilize minimal suites of control effectors. Because of this, failure of just one effector can often have devastating effects on the vehicle's ability to generate the required commanded moments. If it is not physically possible to recover full control effectiveness in all axes, then the vehicle will be operating in a reduced capacity and adaptation of the guidance-feedback gains or architecture will be needed. In some cases, control reconfiguration and guidance adaptation will be enough to recover the mission. However, for many failure scenarios, desired end conditions of the mission segment [e.g., entry–TAEM (terminal area energy management) interface, or touchdown following the approach/landing flight phase] may be unachievable without altering the trajectory commands that drive the guidance loops. Such reshaped trajectories often result in reduced accelerations in response to the vehicle's degraded maneuvering capabilities. Alternatively, the paths may be retargeted to account for the vehicle's new lift and drag characteristics while meeting certain critical constraints, such as heating constraints in hypersonic flight or specifications on sink rate and vehicle attitude to avoid structural damage at touchdown. Control and guidance reconfiguration play essential roles in failure recovery; however, to maximize the level of fault tolerance, these critical elements must be integrated with on-line trajectory command reshaping.

The next section gives some background concerning trajectory reshaping. This is followed by a description of the methodology

and design development. Experimental and flight-test results are presented and conclusions are then drawn from the results.

Background

Onboard trajectory command generation approaches for reusable launch vehicles (RLVs) have recently received considerable attention. This task is currently accomplished with "shuttle-heritage" design approaches. Such methods have worked well in the past and are robust to steady wind variation and navigation error. However, these approaches do not address robustness to drastically changed vehicle dynamics because reconfiguration technologies were not yet available when these command-generation algorithms were developed. Although current shuttle-heritage approaches do have some adaptation attributes, solutions are based largely on extensive off-line analysis, design, testing, and validation. Research is ongoing to develop more flexible, robust approaches to generating and adapting the trajectory commands on line. However, two major challenges have been 1) computational requirements: trajectory generation involving optimization methods is usually computationally quite intensive and often cannot be computed in real time; and 2) convergence guarantees: obtaining viable solutions to the optimization often requires involvement by the designer or engineer through trial and error, which cannot be afforded on line in real time.

A variety of new RLV trajectory-design approaches have recently been proposed. For example, predictor–corrector approaches were presented in Refs. 4–6. In Ref. 4, an entry guidance design was presented with focus on thermal stress constraints. In Refs. 5 and 6, an approach to reentry guidance was presented in which the trajectory solution was obtained prior to the deorbit burn and then followed via a linear–quadratic regulator guidance method during reentry. Here, too, heating constraints were the primary focus. A shooting method used in the procedure was aided by accounting for specific RLV reentry trajectory constraints. An autoland guidance system was presented in Refs. 7 and 8. Here, a two-point boundary value problem was coupled with a propagation technique that relied largely on shuttle-heritage geometric constraints. To increase robustness, solutions were sought that were near the center of the vehicles' control capability. A simplified extension to shuttle-heritage guidance was developed in Refs. 9 and 10 for both the reentry and TAEM flight phases. Another method for entry guidance was developed in Ref. 11. Here, too, the trajectory is computed prior to the deorbit burn, and the solution lies in the middle of the "entry flight corridor" to ensure that it is flyable. The envelope of this corridor is defined by certain specific constraints and conditions: heat rate, normal acceleration, and dynamic pressure limits, and an "equilibrium glide condition."

The trajectory-reshaping approach presented here is known as the optimum-path-to-go (OPTG) methodology and has been successfully applied in simulation to numerous applications, including "smart" munitions, unmanned air vehicles,[12] and several different RLV platforms.[13,14] This approach represents a unique solution to the onboard trajectory-reshaping problem. Unlike most other approaches, this method utilizes solutions precomputed off line. During flight, the trajectory corresponding to the current conditions is chosen, and corrections are made for errors/disturbances. The computational overhead is greatly reduced, as no online prediction/integration is required, and the trajectory database is encoded for efficient interrogation. Further, convergence to a solution is not an issue, as the solution already exists (computed off line).

A notable program in this area of research was the Marshall Space Flight Center's advanced guidance and control (AG&C) program.[15,16] This project was established to examine several control and/or guidance methods to determine approaches that best address NASA's long-term goals. Although not part of the original effort, preliminary developments of the trajectory-reshaping approach presented here were explored through additional AG&C funds.[14] Further technical developments and flight testing of this approach were then conducted under the U.S. Air Force Research Laboratory's Integrated Adaptive Guidance and Control (IAG&C) program, which is the focus of this paper. The demonstration platform for this effort was Boeing's X-40A RLV, an unpiloted test-bed vehicle with autonomous control and guidance. As a technology demonstrator, the X-40A is an 80% scale version of its orbital counterpart, the X-37.

In an earlier program, seven successful drop tests (from a carrier helicopter) of the X-40A were conducted that tested Boeing's original guidance and control systems for the approach/landing flight phase (Fig. 1). The algorithms developed under the IAG&C program were designed for this flight phase in the hopes of eventually repeating similar drop tests, testing the control/guidance reconfiguration and trajectory-reshaping algorithms.

In the IAG&C program, the type of failure was limited to control surfaces that were locked or jammed in place. Single as well as multiple locked-control-surface failures were investigated. The control effectors modeled for the demonstration vehicle are the following:

1) *Left and right ruddervators:* Much like missile fin surfaces, these act both as pitch (asymmetric deflection) and as yaw/roll (symmetric deflection) control effectors.

2) *Left and right flaperons:* These act both as lift generating devices (symmetric deflection) and as ailerons for roll control (asymmetric deflection).

3) *Speedbrake:* This is a surface on the top of the fuselage between the ruddervators and can be deflected upward to 70 deg. It is nominally deflected to approximately 20 deg so that the velocity can be controlled in both directions (reduce deflection to speed up, increase deflection to slow down). Under certain failure scenarios, the speedbrake can be employed to produce some pitching moment.

4) *Body flap:* This surface is hinged at the aft of the fuselage, similar in configuration to a space shuttle's body flap. Under nominal operations, this surface is used for pitch trim control, but it may be used for active pitch control under failures.

Note that although the X-40A is modeled here to include a speedbrake and body flap, the actual vehicle is not equipped with these devices. However, the X-37 is equipped with these effectors. The actuator models and the influence of these effectors on the X-40A vehicle dynamics are drawn from the X-37 simulation model. These additional control effectors are included in the demonstration vehicle to allow more varied failure and reconfiguration studies and to facilitate transition to the X-37 program.

For risk-reduction purposes, the algorithms were flight tested using the General Dynamics total in-flight simulator (TIFS) research aircraft, which simulated the X-40A dynamics (Fig. 2).

Fig. 1 Drop test of Boeing's X-40A RLV.

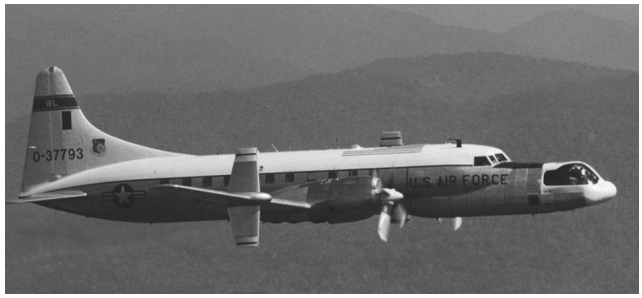

Fig. 2 General Dynamics total in-flight simulator (TIFS) research aircraft.

Note from Fig. 2 that the TIFS has two pilot stations, one farther aft and the other at the extended nose of the vehicle. The former is for the safety pilots, who may take control of the vehicle during the course of the experiment, if necessary, and the latter for the test pilots when the study involves piloted aircraft (not used in the IAG&C program). The TIFS is also equipped with direct lift flaps, which can deflect up or down, and when ganged with the elevators, can produced direct lift forces, absent any pitching moments. Last, note the large side-force surfaces approximately midspan on the wings of the aircraft. In the IAG&C program, these were symmetrically deployed outward to add a large amount of drag to the vehicle to approximate the low L/D ratio of the X-40A.

System Architecture

In the most general case, reconfiguration capabilities for RLVs may employ all of the following elements:

1) *Failure identification:* Because the focus of the IAG&C program was on integration of the inner and outer loop systems with trajectory reshaping, the failures were assumed to be identified using an integrated vehicle health monitoring system, or a "smart" actuation system. This assumption could easily be relaxed through the use of on-line system identification techniques that have previously been flight-demonstrated.[17]

2) *Inner-loop control reconfiguration:* The control system uses measured/identified information regarding the failure to reconfigure the architecture, adapt the feedback gains, or reallocate effector tasks—usually in some optimal fashion. The reconfigurable control system is tasked to maintain attitude stability and recover guidance command following performance to the extent possible.

3) *Outer-loop guidance adaptation:* The adaptive guidance system may also employ some type of identification procedure to "recognize" performance degradations of the inner-closed-loop system and adjust the guidance feedback gains accordingly. The adaptive guidance system is tasked to maintain flight-path stability and recover trajectory command following performance to the extent possible.

4) *Onboard trajectory command reshaping:* Under severe failure conditions, it may also be necessary (or simply prudent for robustness considerations) to adapt the trajectory commands. The trajectory-reshaping algorithm is tasked to find a retargeted trajectory that requires only forces and moments that can be achieved by the degraded vehicle. Further, for RLVs that are both unpowered and unmanned, energy management is critical for a safe landing. If the dynamic and/or aerodynamic characteristics of the vehicle have changed due to some type of subsystem failure or vehicle damage, then the energy-management task may have changed, requiring a retargeted commanded trajectory.

Figure 3 presents the overall system architecture for the elements just described. As indicated in the figure, both measurements and critical parameters are supplied to the OPTG trajectory-reshaping algorithm and the adaptive guidance laws. Following an anomalous event such as a control-effector failure, critical parameters are those measurable (or reconstructable) quantities that convey the key aspects of the vehicle's performance degradations to the outer-loop guidance-reconfiguration or trajectory-reshaping functions. For the current study, the critical parameter for guidance-law adaptation is the inner-loop controller's reference-model bandwidth (a design parameter), which is used to schedule outer-loop guidance gains.

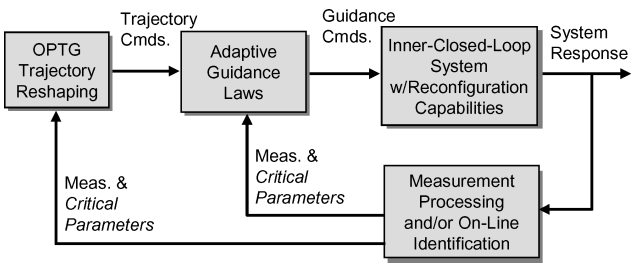

Fig. 3 System architecture with trajectory reshaping.

The critical parameters for trajectory reshaping are upper and lower bounds on achievable lift and drag forces. These are identified on line using a piecewise linear programming (PLP) algorithm, further discussed in Ref. 18. Analogous to the "attainable moment set," often discussed in inner-loop attitude-control studies, the lift and drag bounds characterize the "attainable force set"—important for the outer-loop guidance problem. Lift and drag bounds parameterize all possible failures and anomalous conditions that affect the vehicle's ability to generate these forces. For example, the commanded trajectory may be reshaped in the same fashion for an unforeseen headwind as it would for a particular locked control surface that would add approximately the same amount of additional axial force.

Reconfigurable Control

The purpose of the inner-loop control system is to track body-frame angular-velocity-vector commands. The inner-loop control architecture developed in this work consists of three major components: a dynamic inversion-control law, a control-allocation algorithm, and an explicit model-following system.

Dynamic Inversion

The goal of dynamic inversion in flight control is to cancel the wing–body moments with control moments so that the vehicle can accurately track angular-rate commands generated by the guidance law. The dynamics of the body-frame angular-velocity vector for a lifting body can be written as

$$\dot{\boldsymbol{\omega}} = \boldsymbol{f}(\boldsymbol{\omega}, \boldsymbol{P}) + \boldsymbol{g}(\boldsymbol{P}, \boldsymbol{\delta}) \tag{1}$$

where $\boldsymbol{\omega} = [p \;\; q \;\; r]^T$, and p, q, and r are the rolling, pitching, and yawing rates, respectively; $\boldsymbol{P}$ denotes measurable or estimable quantities that influence the body-frame states, and $\boldsymbol{\delta} = [\delta_1 \;\; \delta_2 \;\; \cdots \;\; \delta_n]$ is a vector of control surface deflections. The vector $\boldsymbol{P}$ contains variables such as angle of attack, sideslip, Mach number, and vehicle mass properties. The term $\boldsymbol{g}(\boldsymbol{P}, \boldsymbol{\delta})$ includes the control-dependent angular accelerations, and the term $\boldsymbol{f}(\boldsymbol{\omega}, \boldsymbol{P})$ describes angular accelerations that are due to the base vehicle's (wing–body) aerodynamic properties. Because the approach and landing flight phase of the X-40A is accomplished dead-stick, it is assumed that the mass properties of the vehicle under consideration are constant. Thus, the time derivative of the inertia matrix can be set to zero; that is, $\dot{\boldsymbol{I}} = \boldsymbol{0}$. (A discrete change in mass properties occurs when the landing gear is deployed; however, this change is small and is therefore ignored in this development.) Under the above assumptions, Eq. (1) can be written as[19]

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}[\boldsymbol{G}_B(\boldsymbol{\omega}, \boldsymbol{P}, \boldsymbol{\delta}) - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}] \tag{2}$$

where

$$\begin{aligned}\boldsymbol{G}_B(\boldsymbol{\omega}, \boldsymbol{P}, \boldsymbol{\delta}) &= \boldsymbol{G}_{WB}(\boldsymbol{\omega}, \boldsymbol{P}) + \boldsymbol{G}_\delta(\boldsymbol{P}, \boldsymbol{\delta}) \\ &= [L \quad M \quad N]^T_{WB} + [L \quad M \quad N]^T_\delta\end{aligned} \tag{3}$$

In Eqs. (2) and (3), $\boldsymbol{I}$ is the inertia matrix, L, M, and N are the rolling, pitching, and yawing moments, $\boldsymbol{G}_{WB}(\boldsymbol{\omega}, \boldsymbol{P})$ is the moment generated by the base aerodynamic system (wing–body system), and $\boldsymbol{G}_\delta(\boldsymbol{P}, \boldsymbol{\delta})$ is the total moment vector produced by the control effectors. Therefore,

$$\boldsymbol{f}(\boldsymbol{\omega}, \boldsymbol{P}) = \boldsymbol{I}^{-1}[\boldsymbol{G}_{WB}(\boldsymbol{\omega}, \boldsymbol{P}) - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}], \qquad \boldsymbol{g}(\boldsymbol{P}, \boldsymbol{\delta}) = \boldsymbol{I}^{-1}\boldsymbol{G}_\delta(\boldsymbol{P}, \boldsymbol{\delta}) \tag{4}$$

To utilize a linear control allocator, it is necessary that the control-dependent portion of the model be linear in the controls. Hence, an affine approximation is developed such that

$$\boldsymbol{G}_\delta(\boldsymbol{P}, \boldsymbol{\delta}) \approx \tilde{\boldsymbol{G}}_\delta(\boldsymbol{P})\boldsymbol{\delta} + \boldsymbol{\varepsilon}(\boldsymbol{P}, \boldsymbol{\delta}) \tag{5}$$

The term $\boldsymbol{\varepsilon}(\boldsymbol{P}, \boldsymbol{\delta})$ is an intercept term[20] for the body-axis angular accelerations, which is used to improve the accuracy of the linear control-allocation algorithm when the control-induced moments are

nonlinear functions of effector deflection. Using Eqs. (1), (4), and (5), the model used for the design of the dynamic inversion control law becomes

$$\dot{\boldsymbol{\omega}} = \boldsymbol{f}(\boldsymbol{\omega}, \boldsymbol{P}) + \boldsymbol{I}^{-1}\tilde{\boldsymbol{G}}_{\delta}(\boldsymbol{P})\boldsymbol{\delta} + \boldsymbol{I}^{-1}\boldsymbol{\varepsilon}(\boldsymbol{P}, \boldsymbol{\delta}) \tag{6}$$

The objective is to find a control law that provides direct control over $\dot{\boldsymbol{\omega}}$, so that $\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_{\text{des}}$. Hence, the inverse control law must satisfy

$$\dot{\boldsymbol{\omega}}_{\text{des}} - \boldsymbol{f}(\boldsymbol{\omega}, \boldsymbol{P}) - \boldsymbol{I}^{-1}\boldsymbol{\varepsilon}(\boldsymbol{P}, \boldsymbol{\delta}) = \boldsymbol{I}^{-1}\tilde{\boldsymbol{G}}_{\delta}(\boldsymbol{P})\boldsymbol{\delta} \tag{7}$$

Equation (7) provides the dynamic inversion-control law for the body-frame angular velocity vector. To solve Eq. (7) for $\boldsymbol{\delta}$, one must solve a control-allocation problem.

Control Allocation

This element of the control system provides the means for control reconfiguration. Standard fixed-gain control methods will utilize each control effector available for the same purpose, regardless of changes to the system. Control allocation, however, determines the appropriate use of all effectors to follow all commands in some optimal sense. Therefore, if one or more control effectors fail, the control system will reallocate (reconfigure) the utilization of the remaining effectors in order to achieve the desired commands. For the vehicle under consideration, there are three controlled variables, namely, roll, pitch, and yaw accelerations, whereas there are six control surfaces. When the number of control effectors exceeds the number of controlled variables, viable command solutions can be achieved in different ways. Therefore, the control-allocation algorithm is used to provide unique solutions to such problems.[21–23] In this work the control allocator is further utilized to generate effector commands that avoid actuator rate or position saturation. Here, the control allocation scheme uses the mixed-optimization linear programming technique of Bodson.[23]

To begin this development, let the left-hand side of Eq. (7) be defined as $\boldsymbol{d}_{\text{des}}$ and denote the right-hand side of Eq. (7) as $\boldsymbol{B}\boldsymbol{\delta}$. Here, the elements of $\boldsymbol{d}_{\text{des}}$ are the body-axis accelerations that must be produced by the control effectors and $\boldsymbol{B}$ is the control effectiveness matrix, defined as

$$\boldsymbol{B} = \boldsymbol{I}^{-1}\tilde{\boldsymbol{G}}_{\delta}(\boldsymbol{P}) = \boldsymbol{I}^{-1}\begin{bmatrix} \dfrac{\partial L}{\partial \delta_1} & \dfrac{\partial L}{\partial \delta_2} & \cdots & \dfrac{\partial L}{\partial \delta_n} \\ \dfrac{\partial M}{\partial \delta_1} & \dfrac{\partial M}{\partial \delta_2} & \cdots & \dfrac{\partial M}{\partial \delta_n} \\ \dfrac{\partial N}{\partial \delta_1} & \dfrac{\partial N}{\partial \delta_2} & \cdots & \dfrac{\partial N}{\partial \delta_n} \end{bmatrix} \tag{8}$$

The following mixed-optimization problem can be posed to solve the control allocation problem:

$$\min_{\delta} J_M = \min_{\delta}(\|\boldsymbol{B}\boldsymbol{\delta} - \boldsymbol{d}_{\text{des}}\|_1 + \lambda\|\boldsymbol{W}_{\delta}(\boldsymbol{\delta} - \boldsymbol{\delta}_p)\|_1) \tag{9}$$

subject to

$$\underline{\boldsymbol{\delta}} \le \boldsymbol{\delta} \le \bar{\boldsymbol{\delta}} \tag{10}$$

In Eq. (9), J_M is the performance index for the mixed-optimization problem. The 1-norm performance index is selected so that linear programming techniques can be used to solve the problem.[23] The vector $\boldsymbol{\delta}_p$ in Eq. (9) is a preference vector. The matrix $\boldsymbol{W}_{\delta}$ is used to weight the importance of driving each control effector to its preferred value. The parameter λ is used to weight the error and control minimization problems. For this work, it was determined that $\lambda = 0.01$ provides good error minimization while still driving the control effectors to the preferred values when sufficient control authority exists. In Eq. (10), $\underline{\boldsymbol{\delta}}$, $\bar{\boldsymbol{\delta}}$ are the most restrictive lower and upper limits on the control effectors, respectively, defined as

$$\bar{\boldsymbol{\delta}} = \min(\boldsymbol{\delta}_U, \boldsymbol{\delta} + \dot{\boldsymbol{\delta}}_{\max}\Delta t), \qquad \underline{\boldsymbol{\delta}} = \max(\boldsymbol{\delta}_L, \boldsymbol{\delta} - \dot{\boldsymbol{\delta}}_{\max}\Delta t) \tag{11}$$

where $\boldsymbol{\delta}_L$, $\boldsymbol{\delta}_U$ are the lower and upper position limits, $\boldsymbol{\delta}$ is the last control-effector command from the control-allocation algorithm, $\dot{\boldsymbol{\delta}}_{\max}$ is a vector of rate limits, and Δt is the time step or control-update rate.

Attention is now turned to the preference vector $\boldsymbol{\delta}_p$ in Eq. (9). When adequate control authority exists to drive the norm of the angular acceleration errors to a sufficiently small value, the allocation algorithm will attempt to minimize the difference between the actual control deflections and $\boldsymbol{\delta}_p$. The preference vector is taken to be the pseudoinverse solution such that

$$\boldsymbol{\delta}_p = -\boldsymbol{c} + \boldsymbol{W}^{-1}\boldsymbol{B}^T(\boldsymbol{B}\boldsymbol{W}^{-1}\boldsymbol{B}^T)^{-1}[\boldsymbol{d}_{\text{des}} + \boldsymbol{B}\boldsymbol{c}] \tag{12}$$

where $\boldsymbol{c}$ is an offset vector and $\boldsymbol{W}$ is a diagonal weighting matrix of the form

$$\boldsymbol{W} = \text{diag}\left[W_{\delta_{\text{RF}}}, W_{\delta_{\text{LF}}}, W_{\delta_{\text{RR}}}, W_{\delta_{\text{LR}}}, W_{\delta_{\text{SB}}}, W_{\delta_{\text{BF}}}\right] \tag{13}$$

where the individual weighting elements are for the right flaperon (RF), left flaperon (LF), right ruddervator (RR), left ruddervator (LR), speedbrake (SB), and bodyflap (BF). The elements of the offset vector $\boldsymbol{c}$ are all zero except those corresponding to locked control surfaces. If a control effector is locked, then the corresponding entry in $\boldsymbol{c}$ is set to the negative of the locked location. Using this preference vector allows one to analytically represent the control allocator in a robustness analysis of the system that is valid as long as no single axis is saturated and the commanded accelerations are feasible.

One last point to consider in the control allocation paradigm is the potential for slope reversals in the individual moment–effector relationships on this vehicle. Typically, slope reversals exist near the maximum or minimum values of control surface deflection. This can create problems when the actuator moves into nonmonotonic regions of the moment-deflection curves, where it is possible for the effector to be limited to an unnecessarily restrictive range of positions. This phenomenon is a direct result of the use of a linear control allocator. Fortunately, this situation can be remedied by ensuring that the moment curve is monotonic throughout the control space. This can be accomplished by preprocessing the moment curve data to eliminate control slope reversals by making the slopes asymptotically approach zero.

In summary, the above formulation solves the angular acceleration error minimization problem and, if sufficient control authority exists, the difference between the control effector positions and a preferred set of effector positions is minimized (control minimization problem).

Explicit Model Following

An explicit model-following scheme is employed so that the inner-closed-loop system exhibits decoupled first-order responses to body-axis angular rate commands. This is achieved by shaping the closed-loop responses and compensating for modeling error in the dynamic inversion-control law and control allocator. The explicit reference model is described by

$$\frac{\omega_m(s)}{\omega_{\text{des}}(s)} = \frac{K_{\text{bw}}}{(s + K_{\text{bw}})} \tag{14}$$

where ω_m denotes either the desired roll, pitch, or yaw rate response of the explicit model and ω_{des} denotes the angular velocity command from the guidance law. The term K_{bw} defines the nominal bandwidth of the desired dynamics. The system is designed to provide perfect tracking of the reference model when the dynamic inversion is perfect. Because this is never the case in practice, error compensation elements (proportional, integral, and derivative channels) are used to mitigate the effects of inversion error. If the inversion is perfect, then the controlled element from the point of view of the explicit model following structure is a simple integrator. From block-diagram algebra (see Fig. 4), one can see that if the controlled element is a simple integrator; that is, if $\omega(s) = (1/s)\dot{\omega}_{\text{des}}(s)$, then the $\omega(s)/\omega_m(s)$ transfer function is given by

$$\frac{\omega(s)}{\omega_m(s)} = \frac{(K_d + K_{\text{ff}})s^2 + K_P s + K_I}{(K_d + 1)s^2 + K_P s + K_I} \tag{15}$$

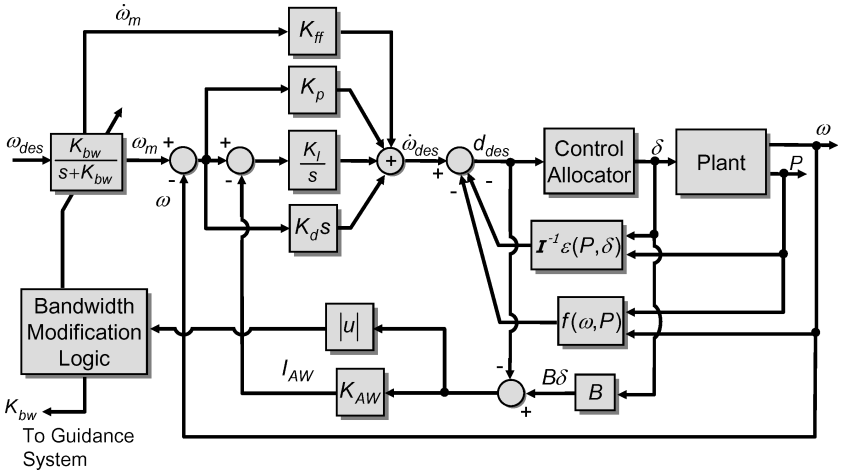

Fig. 4 Inner-loop block diagram.

Table 1 Control-system parameters for explicit model-following control system

Roll	Pitch	Yaw
$K_{\mathrm{bw}_P} = 5$	$K_{\mathrm{bw}_Q} = 3$	$K_{\mathrm{bw}_R} = 5$
$K_{P_P} = 7$	$K_{P_Q} = 10$	$K_{P_R} = 7$
$K_{I_P} = 20$	$K_{I_Q} = 10$	$K_{I_R} = 20$
$K_{D_P} = 0.1$	$K_{D_Q} = 0.1$	$K_{D_R} = 0.1$
$K_{\mathrm{ff}_P} = 1$	$K_{\mathrm{ff}_Q} = 1$	$K_{\mathrm{ff}_R} = 1$

which, when $K_{\mathrm{ff}} = 1$, results in a double stable pole–zero cancellation with appropriate choices of K_P, K_I, and K_d. Then

$$\frac{\omega(s)}{\omega_{\mathrm{des}}(s)} = \frac{K_{\mathrm{bw}}}{(s + K_{\mathrm{bw}})} \tag{16}$$

and perfect model following is achieved.

When the inversion is not perfect, the proportional–integral–derivative network attempts to drive the reference model tracking error to zero. Table 1 shows the gains for each channel of the explicit model-following system. These gains were selected to provide acceptable tracking performance for pulse train commands in all channels.

Integrator Antiwindup and Reference Model Bandwidth Attenuation

Axis saturation occurs when all control power has been exhausted in one or more axes. (Note that control-effector saturation is a necessary, but not sufficient, condition for the occurrence of axis saturation. Control-effector saturation results when one or more control surfaces encounter a rate or position limit.) Axis saturation can be detected through an analysis of the control allocation inputs and outputs. When at least one component of $\boldsymbol{B\delta} - \boldsymbol{d}_{\mathrm{des}} \neq 0$, then axis saturation has occurred. In order to prevent the integrator in the explicit model following prefilter from attempting to cancel tracking errors caused by axis saturation, an integrator antiwindup law is used to reduce the magnitude of the input to the integrator. The integrator antiwindup vector used in this design is given by

$$\boldsymbol{I}_{\mathrm{AW}} = \boldsymbol{K}_{\mathrm{AW}}(\boldsymbol{B\delta} - \boldsymbol{d}_{\mathrm{des}}) \tag{17}$$

where $\boldsymbol{K}_{\mathrm{AW}}$ is a gain. The integrator antiwindup compensation scheme operates on the difference between the angular acceleration estimated by the model used by the control allocator, $\boldsymbol{B\delta}$, and the desired angular acceleration, $\boldsymbol{d}_{\mathrm{des}}$. If no axes are saturated, then $\boldsymbol{B\delta} - \boldsymbol{d}_{\mathrm{des}} = 0$ and the control system operates normally. When $\boldsymbol{B\delta} - \boldsymbol{d}_{\mathrm{des}} \neq 0$, the state of the prefilter integrator is reduced by the antiwindup signal.

The inner-loop control system exhibits degraded performance when an axis saturates and will no longer be capable of tracking nominal commands. When this occurs, the reference-model bandwidth, K_{bw}, is reduced to avoid overdriving the actuators. The law used to reduce the inner-loop bandwidth is as follows: for all three axes, the input to the reduction law is $|\boldsymbol{B\delta} - \boldsymbol{d}_{\mathrm{des}}|$ and this is passed through a saturation block with a lower limit of 0 and an upper limit of 1. Hence, if the difference between $\boldsymbol{B\delta}$ and $\boldsymbol{d}_{\mathrm{des}}$ is small, a zero is output from each saturation block, whereas if the error is large, the output of one or more saturation blocks will be 1. If we denote the output of the saturation blocks for each of the axes by sat_P, sat_Q, and sat_R, then the adapted bandwidths (K_{bw_P}, K_{bw_Q}, K_{bw_R}) are

$$\begin{aligned} K_{\mathrm{bw}_P} &= [(-9/10)\mathrm{sat}_P + 1]K_{\mathrm{bw}_{P\,\mathrm{nom}}} \\ K_{\mathrm{bw}_Q} &= [(-5/6)\mathrm{sat}_Q + 1]K_{\mathrm{bw}_{Q\,\mathrm{nom}}} \\ K_{\mathrm{bw}_R} &= [(-9/10)\mathrm{sat}_R + 1]K_{\mathrm{bw}_{R\,\mathrm{nom}}} \end{aligned} \tag{18}$$

where $K_{\mathrm{bw}_{P\,\mathrm{nom}}}$, $K_{\mathrm{bw}_{Q\,\mathrm{nom}}}$, and $K_{\mathrm{bw}_{R\,\mathrm{nom}}}$ are the nominal bandwidths, which for this vehicle are set to 5, 3, and 5 rad/sec, respectively (as shown in Table 1). Thus, the ranges of bandwidths are 0.5 to 5 rad/sec for the roll and yaw channels and 0.5 to 3 rad/sec for the pitch channel.

Figure 4 shows the complete inner-loop block diagram. Displayed here are the prefilters, dynamic inversion, control allocation, and integrator antiwindup scheme. For implementation in a digital simulation, all continuous time blocks were converted to equivalent discrete time blocks using a Tustin transformation.

Adaptive Guidance

The guidance-law design for the IAG&C program is briefly reviewed here. This system consists of three main elements: 1) a backstepping architecture, used for the majority of the approach/landing mission, 2) an adaptive element, coupled with inner-loop reference-model bandwidth, and 3) a final-flare guidance law. Further details on this design may be found in Ref. 24.

Backstepping Architecture

For longitudinal guidance, a backstepping approach is employed where "pseudocommands" are generated at each loop closure that drives the next most inner loop.[25] These commands are derived by proportional-plus-integral (PI) control coupled with dynamic inversion. For fixed-wing aircraft systems such as the X-40A, a natural loop-closure architecture that allows for backstepping is as follows:

1) altitude loop (outer most loop) to
2) flight-path-angle loop to
3) angle-of-attack loop (inner most loop).

The last loop then generates the pitch-rate command, Q_c, that drives the inner-closed-loop pitch dynamics. This architecture is depicted in Fig. 5. Note that the trajectory-reshaping algorithm delivers altitude, flight-path angle, and flight-path-angle rate commands (H_c,

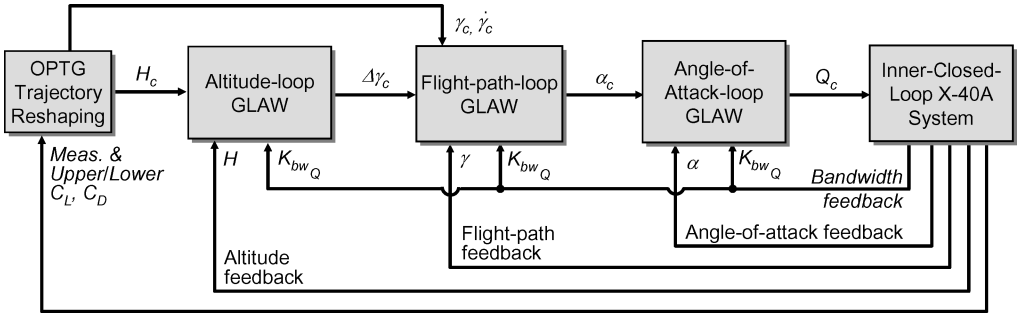

Fig. 5 Backstepping guidance law.

$\gamma_c, \dot{\gamma}_c$) to the guidance law. These parameters define the commanded trajectory. Also see that the reference model pitch-loop bandwidth, K_{bw_Q}, is fed back to each block for gain-adaptation purposes. The guidance laws depicted in each block in Fig. 5 will be discussed in more detail below.

Boeing's original lateral-guidance law was adopted to deliver roll and yaw rate commands to the inner-loop control law. The approach/landing flight condition consists largely of vertical-plane motion; therefore, lateral adaptive-guidance and lateral trajectory-reshaping designs were not pursued in this program.

For the altitude-loop guidance law, the governing equation of motion is

$$\dot{H} = V\sin(\gamma) \tag{19}$$

From dynamic inversion, the flight path command is

$$\gamma_c = \sin^{-1}(\dot{H}_{\mathrm{des}}/V) \tag{20}$$

A corrective flight-path-angle command, or $\Delta\gamma_c$, is generated from a corrective sink-rate command, $\Delta\dot{H}_{\mathrm{des}}$. That is, using Eq. (20), $\Delta\gamma_c$ is defined as

$$\Delta\gamma_c = \sin^{-1}(\Delta\dot{H}_{\mathrm{des}}/V) \triangleq f_\gamma^{-1} \tag{21}$$

Figure 5 indicates that the altitude-loop guidance law delivers this corrective value for flight-path angle instead of a total quantity. The corrective sink-rate command, $\Delta\dot{H}_{\mathrm{des}}$, is derived from PI control on altitude error. The actual implementation of all the algorithms was in the discrete-time domain. However, PI control is presented here in the familiar Laplace domain (i.e., for continuous systems) for clarity. In the Laplace domain, the corrective sink-rate command can be expressed as

$$\Delta\dot{H}_{\mathrm{des}}(s) = (K_H + K_{H_i}/s)(H_c(s) - H(s)) \tag{22}$$

where the altitude command $H_c(s)$ is derived from the OPTG algorithm.

Next, the flight-path-angle loop formulates the appropriate angle-of-attack command. The governing equation of motion here is

$$\dot{\gamma} = L/mV - (g/V)\cos(\gamma) \tag{23}$$

It is assumed here that total vehicle lift, L, can be expressed as follows:

$$L = L_o + L_\alpha\alpha + \{L_\delta\}\boldsymbol{\delta} \tag{24}$$

where L_o is lift at zero angle-of-attack, L_α is the dimensional lift-curve slope, $\{L_\delta\}$ is a row vector of lift increment due to each control surface deflection, and $\boldsymbol{\delta}$ is the vector of control-surface deflection values. From dynamic inversion, the angle-of-attack command is

$$\alpha_c = (mV/L_\alpha)[\dot{\gamma}_{\mathrm{des}} + (g/V)\cos(\gamma) - L'/mV] \triangleq f_\alpha^{-1} \tag{25}$$

where L' is defined as

$$L' = L_o + \{L_\delta\}\boldsymbol{\delta} \tag{26}$$

Table 2 Guidance-loop gain adaptation law

Loop	Adaptation law	Nominal gain value
Altitude	$K_H = (1/15)K_{\mathrm{bw}_Q}$	0.2
Flight-path	$K_\gamma = (1/5)K_{\mathrm{bw}_Q}$	0.6
Angle of attack	$K_\alpha = (5/9)K_{\mathrm{bw}_Q}$	1.67

PI control is again used to derive the desired flight path angle dynamics. Therefore, the desired flight-path-angle rate is defined in the Laplace domain as

$$\dot{\gamma}_{\mathrm{des}}(s) = \dot{\gamma}_c(s) + \left(K_\gamma + K_{\gamma_i}/s\right)(\gamma_c(s) + \Delta\gamma_c(s) - \gamma(s)) \tag{27}$$

where the flight-path-angle and flight-path-angle-rate commands, $\gamma_c(s)$, $\dot{\gamma}_c(s)$, are derived from the OPTG algorithm. Again, the corrective term on the flight-path-angle command, $\Delta\gamma_c(s)$, is derived from the altitude-loop guidance law [see Eq. (21)].

The angle-of-attack loop formulates the appropriate pitch rate command for the inner-loop control law. From $\theta = \alpha + \gamma$, and ignoring lateral-directional influences on the pitch rate so that $Q = \dot{\theta}$, the governing equation of motion here is

$$\dot{\alpha} = -\dot{\gamma} + Q \tag{28}$$

Substituting Eq. (23) for $\dot{\gamma}$, dynamic inversion gives

$$Q_c = \dot{\alpha}_{\mathrm{des}} - (g/V)\cos(\gamma) + L'/mV + (L_\alpha/mV)\alpha \triangleq f_Q^{-1} \tag{29}$$

First-order angle-of-attack tracking is desired; hence, the desired angle-of-attack dynamics are defined as

$$\dot{\alpha}_{\mathrm{des}} = K_\alpha(\alpha_c - \alpha) \tag{30}$$

where α_c is the angle-of-attack command generated from the flight-path loop guidance law (Eq. (25)).

From the preceding equations, the backstepping guidance law is depicted in detail in Fig. 6. In the figure, the functions f_γ^{-1}, f_α^{-1}, and f_Q^{-1} are defined in Eqs. (21), (25), and (29), respectively.

Guidance Law Gain Adaptation

As shown in Fig. 6, the proportional gains within each block of the guidance law are adapted based on the control system's reference-model pitch-loop bandwidth, K_{bw_Q}, (which, recall, is reduced when the inner-loop pitch axis becomes saturated). In particular, the proportional gains are designed to be linear functions of the pitch-axis bandwidth, and this relationship was designed via "loop-shaping" considerations. In this manner, when the pitch-axis bandwidth is reduced, the proportional gains in each of the guidance loops are reduced to maintain stability robustness margins. The adaptation laws used for the flight test program are given in Table 2. It should be noted that nonadaptive systems often contain logic to address control saturation. The approach here is deemed "adaptive" because the feedback gains adapt to changes in the inner-loop bandwidth. Although here such changes are due to saturation, the approach can be generalized to address other reasons for bandwidth changes. For

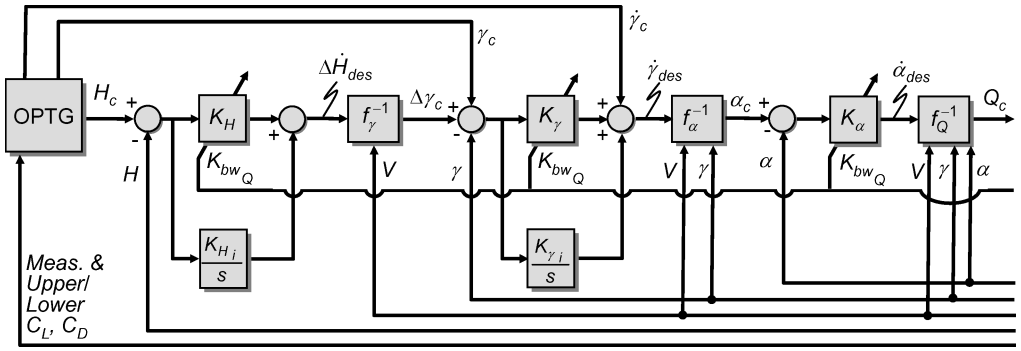

Fig. 6 Backstepping guidance law in detail.

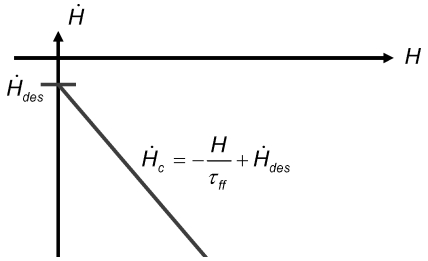

Fig. 7 Altitude versus sink rate profile during final flare guidance.

example, in some vehicles, not all actuators have equivalent bandwidth capabilities, and if high-bandwidth, primary-control effectors fail, slower actuators may have to be utilized in the inner-loop control reconfiguration. In such a case, there will be an overall reduction in inner-loop bandwidth—not due to saturation. The adaptive-guidance scheme presented here would be able to address that class of control failures.

For the integrator gains, antiwindup logic is added to help preserve stability if saturation occurs in either the angle-of-attack loop or the pitch-axis loop of the inner-loop control system. If, for example, the angle-of-attack command exceeds valid bounds defined by the aerodynamic database ($-5 \le \alpha \le 24$ deg), then the integrator states will quickly grow due to tracking errors. A similar situation will occur if the inner loop saturates and can no longer achieve the pitch-rate command. For either of these situations, the integrator states in the altitude and flight-path loops are held at their current values.

Final-Flare Guidance Law

Near touchdown, the primary objective is to arrest the sink rate in preparation for landing. This is accomplished in the final flare guidance law. At an altitude of $H_{\text{ff}} = 150$ ft, the guidance law tracks an exponential sink-rate profile instead of tracking the trajectory provided by the OPTG algorithm. This profile is generated using a linear relationship between altitude and sink rate, as shown in Fig. 7. That is,

$$\dot{H}_c = -H/\tau_{\text{ff}} + \dot{H}_{\text{des}} \tag{31}$$

Here, $\dot{H}_{\text{des}} = -2$ fps is the desired touchdown sink rate.

By adjusting the time constant τ_{ff} the exponential profile can be made more or less aggressive. The time constant is chosen automatically at the start of the final flare to preserve continuity in the sink-rate profile. That is,

$$\tau_{\text{ff}} = H_{\text{ff}} \Big/ \left(\dot{H}_{\text{des}} - \dot{H}_{@H_{\text{ff}}}\right) \tag{32}$$

In addition to preserving continuity, this scheme selects a more or less aggressive time constant depending on the magnitude of the sink rate at H_{ff}.

The sink-rate command is then converted into a flight-path-angle command using the relationship given in Eq. (20). The flight-path-angle-rate command is derived by numerically differentiating the flight-path-angle command.

The final-flare guidance scheme was found to be robust to disturbances and other variations, consistently achieving touchdown sink rates well within the acceptable limits.

Optimum-Path-to-Go Trajectory Reshaping

The four main steps in the OPTG design protocol are 1) optimization problem formulation, 2) off-line trajectory database generation, 3) off-line trajectory database modeling and encoding, and 4) online trajectory reshaping. These steps are covered in detail next.

Optimization Problem Formulation

For the mission segment under study, a trajectory optimization problem is first formulated by defining the objective function, boundaries of admissible initial/final conditions, admissible variations in critical parameters, any particular constraints, and the appropriate governing equations of motion. System states appropriate for the approach/landing flight phase are

V = velocity
γ = flight-path angle
X = downrange
H = altitude

with the corresponding governing equations of motion

$$\begin{aligned} \dot{V} &= -D/m - g\sin(\gamma), & D &= \bar{q}SC_D(\alpha, M) \\ \dot{\gamma} &= L/mV - (g/V)\cos(\gamma), & L &= \bar{q}SC_L(\alpha, M) \\ \dot{X} &= V\cos(\gamma) \\ \dot{H} &= V\sin(\gamma) \end{aligned} \tag{33}$$

Here, $D =$ drag, $L =$ lift, $g =$ acceleration of gravity, $m =$ vehicle mass, $\bar{q} =$ dynamic pressure, $S =$ wing area, C_D and C_L are the drag and lift coefficients, respectively, $\alpha =$ angle of attack, and $M =$ Mach number. To attain additional robustness, common practice in RLV trajectory optimization is to change the independent variable from time to a monotonic parameter.[6] A parameter related to the vehicle's energy state is often used for reentry guidance. An appropriate choice for approach/landing guidance is downrange position, X. To do this, each state equation is multiplied by

$$\frac{\mathrm{d}t}{\mathrm{d}X} = \frac{1}{V\cos(\gamma)} \tag{34}$$

resulting in

$$\begin{aligned} \frac{\mathrm{d}V}{\mathrm{d}X} &= \dot{V}\frac{\mathrm{d}t}{\mathrm{d}X} = V_X = \frac{-D}{mV\cos(\gamma)} - \frac{g}{V}\tan(\gamma) \\ \frac{\mathrm{d}\gamma}{\mathrm{d}X} &= \dot{\gamma}\frac{\mathrm{d}t}{\mathrm{d}X} = \gamma_X = \frac{L}{mV^2\cos(\gamma)} - \frac{g}{V^2} \\ \frac{\mathrm{d}H}{\mathrm{d}X} &= \dot{H}\frac{\mathrm{d}t}{\mathrm{d}X} = H_X = \tan(\gamma) \end{aligned} \tag{35}$$

Equation (34) can be eliminated from consideration because the governing equations are time-invariant and there are no boundary

conditions on time (it is not of concern how long it takes for the RLV to arrive at touchdown). Hence, an added benefit of the new parameterization is that it reduces the dimensionality of the search domain in the optimization problem.

The optimization problem is formulated as a successive quadratic-programming (SQP)[26] problem. For the SQP method, the states and command or decision variables are partitioned into N discrete points. It was found that $N = 100$ is an appropriate choice for approach/landing trajectories. At each point, an approximation is made that generates a solvable quadratic-programming subproblem whose solution is used to form the search direction. The optimization problem is formalized as follows

$$\min_{\substack{C_L \in \mathbb{R}^N \\ X_f \in \mathbb{R}}} J = (\ddot{H}^T W \ddot{H})^{\frac{1}{2}}, \qquad \ddot{H} = f(C_L, X_f) \tag{36}$$

The objective of the cost function is to minimize vertical acceleration, $\ddot{H}$, which is related to the vehicle's normal acceleration, N_z. This allows smoother, less aggressive trajectories, especially beneficial under control-failure scenarios. The weighting matrix, W, is defined as

$$W = \operatorname{diag}\left[1 \quad 1 + \frac{1}{N-1} \quad 1 + \frac{2}{N-1} \quad \cdots \quad 2\right] \tag{37}$$

This weighting scheme emphasizes minimizing accelerations at the end of the trajectory, resulting in less aggressive flares.

The downrange touchdown position, X_f, has an overriding influence on the shape of the trajectories. Therefore, this is included as a decision variable. As expected, for low-energy scenarios, the optimal touchdown point is often near the beginning of the runway, and for high-energy scenarios, it is typically farther down the runway. The other decision variable is the lift-coefficient history, defined as

$$C_L = \left[C_{L_1}, C_{L_2}, \ldots, C_{L_k}, \ldots, C_{L_N}\right]^T \tag{38}$$

Both the lift and drag coefficients are constrained to be within their respective upper and lower bounds, as

$$\begin{aligned} C_{L_{\min}} \le C_{L_i} \le C_{L_{\max}}, \qquad & i = 1, \ldots, N \\ C_{D_{\min}} \le C_{D_i} \le C_{D_{\max}}, \qquad & i = 1, \ldots, N \end{aligned} \tag{39}$$

The lift coefficient is considered to be the single forcing input to the governing equations of motion, because it is assumed that the drag is a linear function of the lift. That is, it is modeled that

$$C_{D_i} = \left(\frac{C_{D_{\max}} - C_{D_{\min}}}{C_{L_{\max}} - C_{L_{\min}}}\right) C_{L_i} + \frac{C_{D_{\min}} C_{L_{\max}} - C_{D_{\max}} C_{L_{\min}}}{C_{L_{\max}} - C_{L_{\min}}} \qquad i = 1, \ldots, N \tag{40}$$

Typically, drag is more of a parabolic function of lift; however, at this stage it is convenient to describe the lift and drag characteristics by two points, their maxima and minima [i.e., $(C_{L_{\min}}, C_{D_{\min}})$, $(C_{L_{\max}}, C_{D_{\max}})$]. Further, this linear approximation was not found to be limiting because the lift and drag coefficients are determined from a trimmed solution, which is not unique. (The particular trimmed solution used here minimized control surface deflections from zero deg for all surfaces except the speedbrake, which was nominally deflected to 35 deg.) That is, in reality, the lift–drag relationship is characterized by a range of parabolas, and the linear relationship used here lies within this range.

Last, the optimization is subject to the following terminal constraints:

$$\begin{aligned} X_{f_{\min}} \le X_f \le X_{f_{\max}}, \qquad & V_f \le V_{f_{\max}} \\ \dot{H}_f \ge \dot{H}_{f_{\min}}, \qquad & H_f = H_{\text{Rnwy}} \end{aligned} \tag{41}$$

The downrange touchdown position is constrained to be between the end of the runway and the point at which the minimum distance required to stop is reached (with margin included). This is defined to be 0 ft $\le X \le$ 10,000 ft (this large range is appropriate for the vehicle under study, because RLVs will typically be targeted to exceptionally long runways or dry lake beds for risk mitigation). For the final velocity, $V_{f_{\max}} = 300$ fps. Much beyond this value, tire blowout can occur. For touchdown sink rate, $\dot{H}_{f_{\min}} = -10$ fps. Damage to the vehicle can occur for sink rates harder than this limit. Last, the final altitude is set to the runway altitude, H_{Rnwy}.

Note that, within the SQP algorithm, the terminal values are found by numerically integrating the system in Eq. (35) from an initial downrange, X_o, to the final touchdown point, X_f, driven by the N decision variables in the "command vector" C_L. In this manner, the state equations are implicitly satisfied even though they are not directly defined as constraints.

Off-Line Trajectory Database Generation

Once the optimization problem is properly formulated, candidate trajectories can readily be obtained. A database of neighboring optimal extremals is generated by successively varying the following:

1) *Initial states:* Understandably, variations in initial conditions will generate slightly reshaped trajectories.

2) *Critical parameters:* Recall that the upper and lower bounds on the lift and drag coefficients, shown in Eq. (39), are defined as the critical parameters. Markedly different trajectory solutions are obtained for variations in these bounds. Such trajectories represent optimal flight paths for the vehicle under failures that alter its lift and drag characteristics.

It is important to note that generation of the trajectory database is not accomplished by generating different trajectories specifically for different control-effector failures. Such a method would be too cumbersome to be a practical approach due to the large (infinite) number of possible failure positions and combinations of multiple control failures. Instead, it was recognized that by their nature, locked control-surface failures change the lift and drag characteristics; however, they also require the unfailed surfaces to be deployed in order to maintain rotational equilibrium, further changing the vehicle's lift and drag characteristics. These perturbations to lift and drag drive the governing equations for flight-path motion [see Eq. (33)]. Therefore, varying the upper and lower bounds on the lift and drag coefficients is a systematic approach to describing the effects of control-surface failures. To develop appropriate variations in lift and drag bounds that would correspond to actual failure conditions, the aerodynamic characteristics for a significant number of failure cases were investigated. Figure 8 presents a plot of C_D versus C_L for the nominal vehicle, and example failure cases. Shown are a speedbrake failure, a double ruddervator failure, and a combination double ruddervator/bodyflap failure. Note that as the severity of the failure increases, the range between minimum and maximum C_D

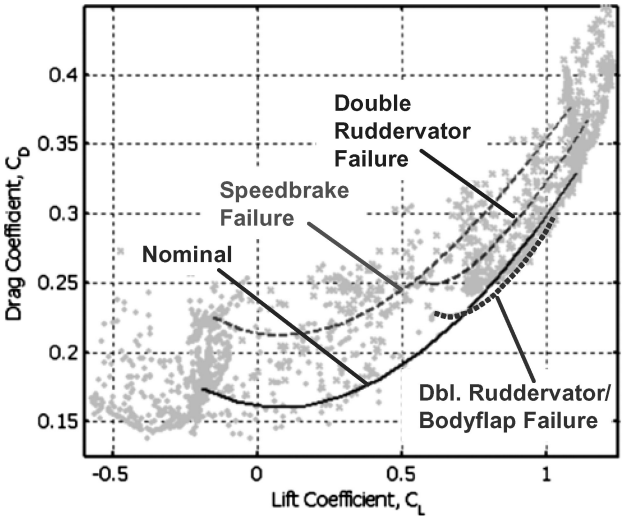

Fig. 8 Aerodynamic characterisitics under failures.

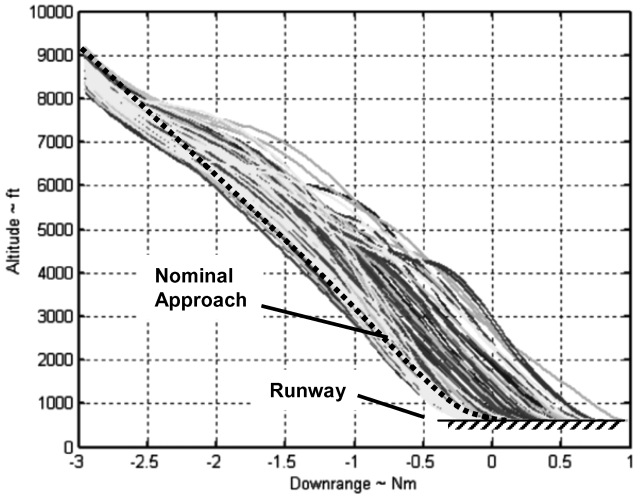

Fig. 9 Partial database of X-40A approach/landing trajectories for variations in lift and drag.

and C_L decreases significantly. Also shown in the background of Fig. 8 are the minimum and maximum (C_L, C_D) points for a multitude of actual failure cases. Data such as these are used to model normally distributed variations in max/min C_L and C_D values that form the set of admissible critical parameters. Last, note that the dependence on Mach number in the lift and drag coefficients was ignored here. For the approach/landing flight region, where Mach number does not significantly change, this assumption is reasonable. However, it is noted that if the OPTG approach is to be applied to, say, the TAEM or reentry flight regions, which cover a wide range in Mach number, then this dependence cannot be ignored. In these cases, the critical parameters must be modeled as functions of Mach number.

It is envisioned that more matured OPTG designs will utilize more "generic" trajectory databases so that the approach is not limited to one specific landing site or one set of scenarios. To this end, it is envisioned that future developments of the OPTG approach will incorporate libraries of trajectory databases, generated by repeating the above procedure for several downrange locations (usually at equally spaced intervals). In this manner, trajectory reshaping will be a continuous process and can address failures/anomalies that occur anywhere in the trajectory. However, for the flight-test program, only one database was generated for variations in vehicle states and lift and drag bounds at the beginning of the approach/landing trajectory. Figure 9 presents a partial database for the X-40A model for successive variations in initial states and lift and drag coefficient bounds. This large volume of trajectory data can be cumbersome to interrogate on line with traditional table-lookup methods. The next step in the OPTG protocol solves this problem.

Off-Line Trajectory Database Modeling and Encoding

This aspect of the OPTG methodology is the most important in terms of enabling on-line use. First, from the trajectory database, the states of the systems are modeled as a set of selected basis functions most appropriate for the mission segment. For the largely two-dimensional vertical-plane approach/landing trajectories, the velocity and altitude states are modeled as polynomials in the downrange, X, via a least-squares curve-fitting routine. That is,

$$V_{\text{cmd}} = a_n X^n + \cdots + a_2 X^2 + a_1 X + a_o$$

$$H_{\text{cmd}} = b_n X^n + \cdots + b_2 X^2 + b_1 X + b_o \tag{42}$$

Polynomials of fifth or sixth order typically gave the "best" fits for the trajectories under study.

A nonlinear function-modeling tool is then used to generate polynomial neural networks (PNNs) that map the initial states and upper/lower bounds on C_D and C_L to the coefficients of the basis functions describing the associated trajectory. PNNs are compositions of Kolmogorov–Gabor (KG) multinomials,[27] or algebraic sums of terms. The near universality of multinomials in representing physical processes, including dynamical systems, has been demonstrated. The KG multinomial, which is given by

$$y = a_0 + \sum_i a_i x_i + \sum_i \sum_j a_{ij} x_i x_j + \sum_i \sum_j \sum_k a_{ijk} x_i x_j x_k + \cdots \tag{43}$$

can model any analytic single-valued transformation. The PNNs are generated through the use of the software package GNOSIS.[28] In GNOSIS, a generic statistical criterion has been developed to govern composition and selection of functions during network synthesis. This criterion takes the form.

$$J = \frac{1}{N} \sum_{i=1}^{N} d(y_i, f(x_i, \hat{g})) + \frac{k}{N} \tag{44}$$

where J is the criterion to be minimized for a given output y, N is the number of input–output pairs in the synthesis database, y_i is the ith value of the output, x_i is the ith value of the input vector in the database, $\hat{g}$ is the estimate of the optimum parameter vector for the network, and k is an information-theoretic measure of the network complexity. The loss or distortion function $d(y_i, \hat{f}_i)$ can take various forms as specified by the analyst, allowing any number of useful regression paradigms (e.g., least squares, ridge regression, logistic regression) to be extended to nonlinear structure-learning neural networks. This provides substantially improved performance over algorithms that minimize squared error alone.

In summary, the PNN mappings essentially describe the trajectory database in an efficient, compact manner. Last, note that the complete off-line procedure, from generating the trajectory database to modeling the trajectories as polynomials and then encoding the information as PNNs, is largely automated.

Online Trajectory Reshaping

The final step in the OPTG approach is performed during flight and entails PNN interrogation and trajectory-command generation. For the IAG&C flight test program, measurements of the initial vehicle states and estimates of the critical parameters were used to interrogate the onboard networks to compute an appropriate set of basis function coefficients. Estimates of the critical parameters were determined on line by posing the following set of optimization problems:

$$\begin{aligned} &\max_{\delta \in \mathbb{R}^N, \alpha \in \mathbb{R}} J = C_L(\delta, \alpha), \qquad \max_{\delta \in \mathbb{R}^N, \alpha \in \mathbb{R}} J = C_D(\delta, \alpha) \\ &\min_{\delta \in \mathbb{R}^N, \alpha \in \mathbb{R}} J = C_L(\delta, \alpha), \qquad \min_{\delta \in \mathbb{R}^N, \alpha \in \mathbb{R}} J = C_D(\delta, \alpha) \end{aligned} \tag{45}$$

subject to 1. $C_m(\delta, \alpha) = 0$ (pitch trim)
2. upper and lower limits on δ

Here, the decision variables are the vector of control surface deflections, δ, and the angle of attack, α. Hence, at the start of each run, the optimization routine finds the maximum and minimum achievable lift and drag coefficients over 1) the range of valid angles of attack, defined to be the range over which the vehicle can be trimmed in pitch, and 2) the achievable ranges in control surface deflections (current control surface position and deflection range characteristics are required for the inner-loop controller and are therefore available for this step—it is assumed here that this information is obtained from a vehicle health monitoring system). Here, any locked control surface would clearly affect the optimization results.

Note that the optimization is subject to the vehicle being in pitch trim. Again, the lift and drag models used for the trajectory database generation are pitch-trim models. Therefore, to correspond to the database, the optimization is also constrained to be in pitch trim. In flight, this gives some controllability margin in that the OPTG algorithm is supplying a commanded trajectory that the vehicle should be able to fly while in trim. If needed, the vehicle should be capable of generating lift and drag forces beyond the upper and lower bounds given by the trim models.

The objective here is to obtain estimates of the lift and drag coefficients on board in (near) real time. To achieve solutions as rapidly and accurately as possible, a piecewise linear programming algorithm[18] is used to solve the set of optimization problems given in Eq. (45). The PLP approach is an optimization method that allows nonlinear programming problems to be solved as linear programming problems by approximating each nonlinear term as a piecewise linear function. In order to accomplish this, the nonlinear cost function and constraints must be in separable form. For example, the pitching moment constraint must be described as follows:

$$C_m(\delta,\alpha) = C_{m\_\delta_{\text{flaps}}}(\delta_{\text{flaps}}) + C_{m\_\text{rud}}(\delta_{\text{rud}}) + \cdots + C_{m\_\text{bf}}(\delta_{\text{bf}}) + C_{m_\alpha}(\alpha) \tag{46}$$

where each term is a (possibly nonlinear) function of a single decision variable. Then, each nonlinear term in the separable programming problem is approximated as a piecewise linear function. This is accomplished by determining the valid range for each variable and then dividing the variable into an appropriate number of linear segments. The decision variables are then redefined to be "weights" associated with each breakpoint for each piecewise linear function. Additional constraints on these weights are imposed to ensure that no segments of each piecewise linear function are skipped in the search for the optimal solution. More details of the PLP approach can be found in Ref. 18. Testing of the PLP approach indicated that this method can rapidly produce accurate estimates of the bounds on lift and drag coefficients.

Sensor noise and atmospheric disturbances, such as gusts or unpredicted winds, are difficult to include in a trajectory database and thus would not be represented in the PNN models. To be robust to such disturbances and errors, the PNNs are augmented with an on-line coefficient correction algorithm to ensure that the current and final conditions are met. A constrained least-squares approach is employed for the coefficient correction algorithm, where the objectives are to 1) correct the coefficients to eliminate errors in the current and final boundary conditions, and 2) preserve the original shape of the polynomials to the extent possible. The cost function reflecting these two objectives is

$$J = \tfrac{1}{2}\left\|\sqrt{R}(Y - \Phi\tilde{\theta})\right\|_2 + \tfrac{1}{2}\left\|\sqrt{W}(\tilde{\theta} - \theta_o)\right\|_2 \tag{47}$$

where Y is the polynomial output at the current and final boundaries, Φ is the regressor at the current and final boundaries, $\tilde{\theta}$ is the corrected coefficients, and θ_o is the uncorrected coefficients given by the PNNs. R and W are relative weighting matrices used for tuning the solution. The solution is given by

$$\tilde{\theta} = [\Phi^T R\Phi + W + \widehat{R}]^{-1}[\Phi^T RY + W\theta_o] \tag{48}$$

where $\widehat{R} = 10^{-6}I$ is a regularization matrix used to avoid singularities in the solution. For velocity,

$$Y = \begin{bmatrix} V_j \\ V_f \end{bmatrix}, \qquad \Phi = \begin{bmatrix} X_j^n, \ldots, X_j^2, X_j, 1 \\ X_f^n, \ldots, X_f^2, X_f, 1 \end{bmatrix} \tag{49}$$

However, for altitude, it is also desired to match the slope at the current and final boundaries [slope $= \partial H/\partial X = \tan(\gamma)$; see Eq. (35)]. Therefore, in this case,

$$Y = \begin{bmatrix} H_j \\ \tan(\gamma_j) \\ H_f \\ \tan(\gamma_f) \end{bmatrix}, \qquad \Phi = \begin{bmatrix} X_j^n, \ldots, X_j^2, X_j, 1 \\ nX_j^{n-1}, \ldots, 2X_j, 1 \\ X_f^n, \ldots, X_f^2, X_f, 1 \\ nX_f^{n-1}, \ldots, 2X_f, 1 \end{bmatrix} \tag{50}$$

With this, the corrected velocity and altitude commands can be expressed as

$$V^*_{\text{cmd}} = a_n^* X^n + \cdots + a_2^* X^2 + a_1^* X + a_o^*$$

$$H^*_{\text{cmd}} = b_n^* X^n + \cdots + b_2^* X^2 + b_1^* X + b_o^* \tag{51}$$

Flight-path-angle commands are subsequently derived from the equations of motion:

$$\gamma^*_{\text{cmd}} = \tan^{-1}\left(\frac{\partial H^*_{\text{cmd}}}{\partial X}\right) \tag{52}$$

Flight-path-angle-rate commands are derived numerically as follows:

$$\dot{X}^*_{\text{cmd}} = V^*_{\text{cmd}}\cos\left(\gamma^*_{\text{cmd}}\right), \qquad X^+ = X + \dot{X}^*_{\text{cmd}}\,\mathrm{d}t$$

$$\dot{\gamma}^*_{\text{cmd}} = \left(\gamma^*_{\text{cmd}}(X^+) - \gamma^*_{\text{cmd}}(X)\right)/\mathrm{d}t \tag{53}$$

Again, H^*_{cmd}, γ^*_{cmd}, and $\dot{\gamma}^*_{\text{cmd}}$ define the commanded trajectory, and these quantities are delivered to the guidance law (see Fig. 5).

Batch Simulation Results

Flight of the X-40A during approach/landing was studied both under nominal conditions (all control surfaces working) and for a significant number of locked control-surface failure cases. Due to programmatic constraints, the focus of failure experiments was limited to locked surfaces, because this was deemed one of the more probable types of failures. Other classes of failures, such as floating, missing, damaged, or partially working surfaces, would have to be investigated in further developments of this approach.

The final phase of batch-simulation experiments was conducted just prior to the final phase of flight tests, and these results are presented in Table 3. These results were obtained from the high-fidelity 6DOF simulation model for the X-40A, developed for the IAG&C flight test program. This simulation includes high-fidelity aerodynamic and actuator models and geodetic, navigation, and measurement-processing modules provided by Boeing. These elements are integrated with the final control, guidance, and trajectory-reshaping designs. At this late stage in the design cycle, study focused on severe-failure cases in which two or more control effectors were locked in place. These represent good candidates for scenarios requiring in-flight trajectory reshaping, having significant effect on lift and drag limits. In the table, BF = body flap, Rud = ruddervator, Flap = flaperon, and SB = speedbrake. For all cases presented in Table 3, both the control-reconfiguration and guidance-gain-adaptation functions are enabled. Without at least these capabilities, the vehicle will fail to achieve a safe landing. Each failure class listed in the table represents a number of specific failures studied, where the control surfaces that were failed were locked in position at certain intervals throughout their deflection ranges. For example, the body flap is failed in increments of 5 deg from its full deflection down to its full deflection up. Deflection of the ruddervators is more sensitive, and these are failed in increments of 1 deg from ±3-deg limits. Similar failure suites are defined for the other control surfaces. A total of 2518 cases were run in this experiment, representing various combinations of failed surfaces and deflections at which the surfaces were locked. Table 3 presents results for the IAG&C system with and without OPTG trajectory reshaping enabled. Listed are the percentages of successful touchdowns, where success is defined by the touchdown criteria listed in Table 4.

The results indicate that the IAG&C system is fairly robust to a wide array of multiple surface failures even without trajectory

Table 3 Summary of failure classes investigated

Class of failure	Number of cases	% success w/o reshaping	% success with reshaping
BF + Rud (both)	89	42	77
BF + Flap (left)	21	57	100
BF + SB	1599	60	100
SB + Rud (both)	639	63	89
SB + Flap (left)	81	21	76
Flap (left) + Rud (both)	89	82	94
Total	2518	59	95

Table 4 Touchdown specifications

Touchdown specification	Failure to meet specs
Sink rate ≥ -10 fps	Landing gear or structural damage
Pitch angle ≤ 15 deg	Tail scrape
Groundspeed ≤ 310 fps	Tire blowout
Downrange $\leq 10{,}000$ ft	Wheel stop off runway

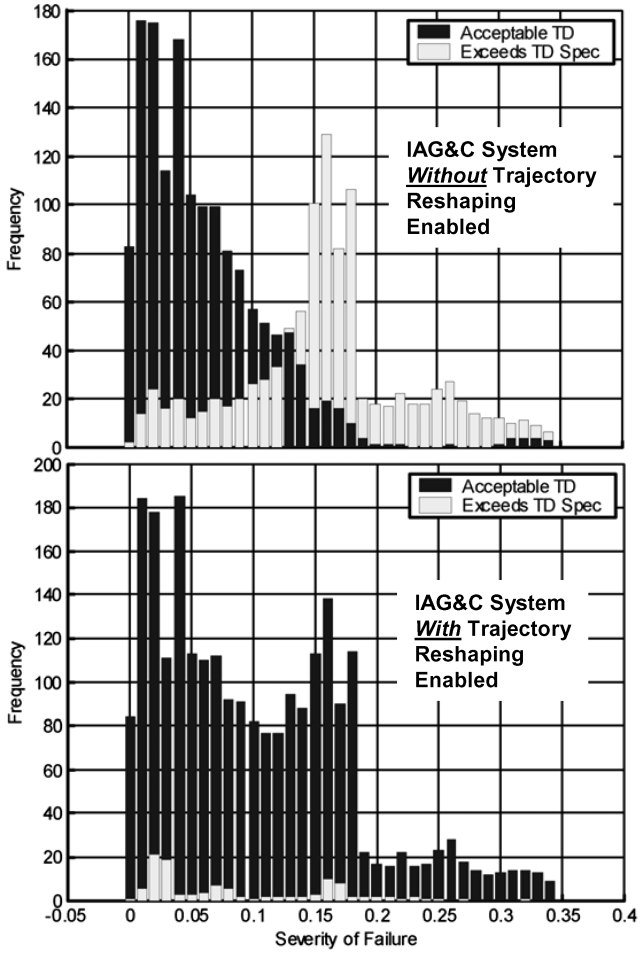

Fig. 10 Histogram summary of batch simulation results for IAG&C system.

reshaping. Approximately 59% of the cases result in successful touchdown conditions for the vehicle tracking the nominal commanded trajectory. The majority of these cases are recovered due to the inner-loop control reconfiguration alone. However, substantial improvement is seen when trajectory reshaping is enabled. Under these cases, approximately 95% of the runs result in successful touchdowns. The 5% of unrecovered cases are failures so severe that loss of the vehicle occurs even with trajectory reshaping. These cases usually involve either a flaperon failure or double rudderva-tor failures in combination with another locked control surface. All remaining control power is allocated simply to maintain vehicle stability, leaving little to no margin for maneuvering the vehicle, especially through the main flare. Under these cases, the vehicle will fly straight into the ground, far short of the runway.

Figure 10 presents these results in histogram form. Plotted is the frequency of occurrence versus a measure of the severity of failure. This measure is defined as the total distance that the critical parameter values are from their respective nominal values:

$$S = \sqrt{f_d\left(C_{L_{\max}}\right) + f_d\left(C_{L_{\min}}\right) + f_d\left(C_{D_{\max}}\right) + f_d\left(C_{D_{\min}}\right)}$$

$$f_d(C_{(\cdot)}) \triangleq \left(C_{(\cdot)} - C^*_{(\cdot)}\right)^2 \tag{54}$$

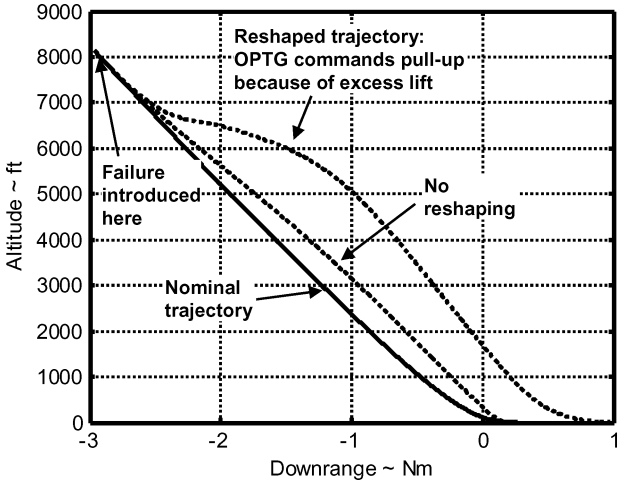

Fig. 11 Trajectory profile for example batch simulation experiment.

where the superscript * represents the nominal value in the function f_d. The top histogram plots the results for the system without trajectory reshaping, and the bottom histogram plots the results for the system with trajectory reshaping. The dark shaded bars represent successful touchdowns, whereas the light shaded bars represent unsuccessful touchdowns. The improvement in recovery performance due to trajectory reshaping is evident. Again, the small number of unrecovered cases in the bottom histogram are those runs in which the vehicle experiences loss of controllability.

Figure 11 presents the altitude profile for an example case study. With no failures, the vehicle is able to follow the nominal commanded trajectory quite accurately. The nominal trajectory is obtained from the OPTG algorithm and is similar to shuttle-heritage approach/landing profiles. The longest portion of such trajectories is made up of a steep glideslope, followed by a main flare to a short, shallow glide slope. A final flare at the end of the trajectory arrests the sink rate just before touchdown. For the failure case study, the body flap is failed at 0 deg and both ruddervators are failed asymmetrically (do not cause roll/yaw) at -3 deg. The failure occurs at the beginning of the trajectory. This is one of the most severe failures from which recovery is physically possible. Shown are results for the system with trajectory-reshaping capabilities provided by the OPTG algorithm and without trajectory-reshaping capabilities. For the latter case, the vehicle attempts to follow the nominal commands. Control reconfiguration and guidance adaptation are active throughout the flight for both cases.

It can be seen that the vehicle under the failure condition without command reshaping cannot follow the nominal commanded trajectory. In contrast, the vehicle with command reshaping can accurately follow the reshaped commanded trajectory. It can be shown that the flaperons are required to deflect trailing edge down to counteract the nose-up pitching produced by the asymmetrically locked ruddervators. This, in turn, adds lift to the trimmed configuration. The OPTG trajectory-reshaping algorithm correctly supplies a commanded trajectory that reflects this added lift, resulting in a trajectory with an immediate pull up in the flight path.

Figure 12 presents the flight-path-angle histories for this case study. Here too, it can be seen that the nominal commanded flight path cannot be followed under the failure condition. However, with command reshaping, only minor tracking errors are noted. The flight-path angles at an altitude of 150 ft are indicated. Recall that it is at this altitude that the final-flare guidance mode begins. Flight paths of approximately -2 deg are targeted at touchdown. It can be seen that for the case with no reshaping, the vehicle is required to turn the flight path from approximately -20 deg to -2 deg in only 150 ft. The vehicle cannot accomplish such an aggressive pull-up maneuver, especially under this severe failure condition. However, for the case with trajectory reshaping, the vehicle has only to reduce its glide slope by approximately 7 deg. This is physically possible and indicates that the OPTG algorithm targets the vehicle to proper

Table 5 Touchdown results for example batch-simulation experiment

Touchdown parameter	*X*-40A TD specs	IAG&C without trajectory reshaping	IAG&C with trajectory reshaping
Sink rate	≥ -10 fps	-24.6 fps	-7.2 fps
Pitch angle	≤ 15 deg	18.1 deg	13.6 deg
Ground speed	≤ 310 fps	303 fps	239 fps
Downrange	$\leq 10{,}000$ ft	1130 ft	5944 ft

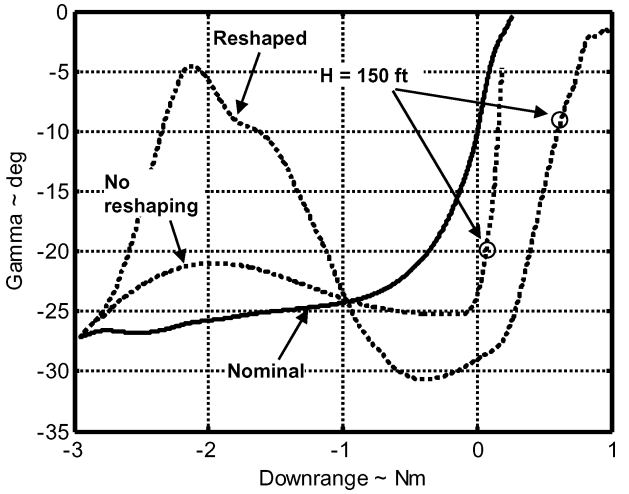

Fig. 12 Flight-path-angle histories for example batch-simulation experiment.

attitude and flight-path states just before touchdown. (Although not shown, the angle-of-attack history for the case with command reshaping is similar in magnitude to the nominal case. For the case with no reshaping, the angle of attack is oscillatory during the main flare, with magnitudes nearing the validity bounds for the aerodynamic model.)

Table 5 presents the touchdown parameters for this case study. It can be seen that the touchdown sink rate for the system without trajectory reshaping is nearly -25 fps. If not completely destroyed, the vehicle would most certainly suffer severe damage. For the system with trajectory reshaping, the touchdown sink rate is well within acceptable limits. Comparing touchdown pitch attitudes, it can be seen that the vehicle would experience tail scrape without trajectory reshaping, but would not with trajectory reshaping. Last, although the touchdown velocities are within specification for both cases, it is much nearer the limit for the case without trajectory reshaping.

Note that Boeing, Huntington Beach, performed extensive Monte Carlo and worst-on-worst dispersion studies in their verification and validation of the algorithms in preparation for flight tests and follow-on studies. Variations studied included 1) initial conditions, 2) winds, 3) sensor errors, 4) atmospheric variations, and 5) aerodynamic modeling errors. Although not presented here, note that the nominal (without failures) IAG&C system passes all worst-on-worst cases. Further, even in severe failure cases, the IAG&C system passes a significant number of worst-on-worst cases, clearly outperforming a nonadaptive G&C design.

Flight Test Results

All of the flight tests were conducted at the Niagara Falls Airport, near the current facilities of General Dynamics at Buffalo–Niagara International Airport in Buffalo, New York. There were three main phases to the flight test program, discussed below.

Calibration flight tests: The first set of flight tests took place over the summer of 2003, and the main objective was to determine the TIFS capabilities in simulating X-40A dynamics. Although the TIFS was used to simulate space shuttle approach/landings in the 1970s, the X-40A flies at a much steeper glideslope (nominally near $\gamma = -27$ deg), and the L/D ratio of the TIFS had to be significantly lowered in order to match the X-40A characteristics. Toward that end, the landing gear was deployed during the flight-test experiments and the side force surfaces were symmetrically deployed trailing-edge-out to add drag to the vehicle. The calibration tests determined how far out the side force surfaces could be deflected before structural limits were reached and severe buffeting occurred. From the results of these tests it was determined that the structural and aerodynamic limits of the TIFS precluded flying the steep flight-path angles of the X-40A.

To solve this problem, the altitude and sink rate parameters passed to the TIFS model-following algorithms during flight were scaled by a factor of 0.3. In this manner, the TIFS vertical profile was 30% of the X-40A from the start of the trajectory to touchdown. For example, if the X-40A began the approach at 10,000 ft altitude (above ground), then the TIFS would start at 3000 ft altitude. Their altitudes would only match at touchdown. The advantage of this methodology was that it allowed the TIFS to be used to flight test the IAG&C system. However, the disadvantage was that sensor errors and vertical winds were magnified by the (inverse) scaling process. The altitude and sink rate parameters fed back from the TIFS to the X-40A model included sensor errors and vertical winds, and these parameters were scaled by a factor of 3.3 ($=1/0.3$). This typically led to larger touchdown sink rates in flight simulations than were observed in batch simulations.

Phase I evaluation flight tests: The first set of evaluation flight tests took place in September through October of 2003. The IAG&C system was first flight-tested during this phase, leading to the final developments of IAG&C technologies, and final modifications of the TIFS modeling following approach. The TIFS simulated autonomous approach/landings of the X-40A under a variety of single and multiple control surface failure experiments. A buildup approach was conducted, first testing failures requiring only inner-loop control reconfiguration to recover, and gradually building up to testing severe failures requiring control/guidance adaptation and trajectory reshaping. Although most touchdowns were simulated at altitude (typically 200 ft), the safety pilots of the TIFS vehicle were comfortable enough with the simulated (unpiloted) system to allow some of the experiments to complete actual touchdowns.

Phase II evaluation flight tests: The final set of flight tests took place in November through December of 2003. The final design of the IAG&C system was flight-tested during this phase of the program. Here, two "baseline" tests were completed in which the trajectory reshaping was disabled. Batch simulations indicated that the vehicle would suffer sever damage; therefore the TIFS needed to land "at altitude" for these baseline runs. For consistency in results, all Phase II tests simulated touchdowns at an altitude of 200 ft.

Typically, one or two nominal runs were first completed for each flight for calibration purposes. In addition to these, there were 40 completed evaluation runs testing 21 different failure combinations. Table 6 presents a summary of the touchdown results for the Phase II evaluation experiments. The runs are listed in chronological order, listing the surfaces failed and the deflection angle of the failures [e.g., L_Flap(-16), SB(55) = Left flaperon failed at -16 deg. (trailing edge up), Speedbrake locked at 55 deg (up); also note that R/L_Rud implies that both right and left ruddervators failed]. These failure combinations were chosen for their extreme trajectory-reshaping characteristics, determined from prior batch-simulation experiments. This table lists the touchdown sink rate and pitch attitude for each run. The touchdown specifications on velocity and downrange position were always met (see Table 4) and are therefore not listed. Those runs considered to have resulted in successful touchdowns are shown in boldface. Although not all of these met the touchdown specifications, they were close enough to be considered successful in light of the model-following scaling issue on altitude and sink rate, discussed earlier. This determination was also based on examination of all the state and control histories for these runs.

There were two main reasons for the unsuccessful runs. First, the Phase II flight tests were plagued by poor weather conditions. Severe winds and turbulence were often encountered, with conditions typically outside original X-40A specifications on acceptable wind conditions. This too was exacerbated by the model-following scaling issue. Second, some trajectory-reshaping cases were so severe

Table 6 Summary of Phase II flight-test touchdown results

Surfaces failed	Touchdown sink rate, fps	Touchdown pitch attitude, deg	Surfaces failed	Touchdown sink rate, fps	Touchdown pitch attitude, deg
L_Flap(−16), SB(55)	**−13.4**	**12.4**	R/L_Rud(−2), SB(70)	−19.2	11.8
L_Flap(−16), SB(55)[a]	−17.0	20.1	**R/L_Rud(−2), SB(60)**	**−10.5**	**9.1**
L_Rud(−4), SB(70)	**−7.2**	**13.7**	R/L_Rud(−4), SB(60)	−26.2	16.0
L_Rud(−4), SB(70)[a]	−21.9	11.5	R/L_Rud(1), SB(60)	+2.0	15.8
SB(70), BF(−20)	**−6.8**	**11.7**	**R/L_Rud(1), SB(70)**	**−5.1**	**13.4**
L_Flap(−16), SB(55)	−20.7	14.8	SB(50), BF(20)	−21.3	15.0
L_Rud(−4), SB(60)	**−6.6**	**17.3**	R/L_Rud(2), SB(70)	−18.4	12.4
SB(60), BF(−20)	**−7.4**	**12.3**	L_Flap(20), SB(60)	−39.2	14.0
R/L_Rud (−4), SB(70)	**−8.8**	**12.2**	**SB(70), BF(20)**	**−13.4**	**18.1**
L_Flap(−16), SB(55)	**−9.8**	**17.8**	**L_Flap(−16), SB(55)**	**−10.4**	**16.0**
SB(50), BF(20)	−17.4	14.1	R/L_Rud(−4), SB(70)	−15.8	14.6
R/L_Rud (2), SB(70)	−26.0	16.3	**R/L_Rud(−4), SB(50)**	**−12.5**	**13.8**
L_Flap(20), SB(60)	−28.3	14.9	**R/L_Rud(−4), SB(60)**	**−10.0**	**15.8**
L_Rud(−4), SB(50)	**−6.6**	**13.8**	R/L_Rud(1), SB(60)	−23.6	19.9
L_Flap(−16), SB(50)	**−4.5**	**13.6**	**R/L_Rud(−2), SB(60)**	**−11.8**	**13.5**
L_Flap(20), SB(65)	**−2.7**	**12.3**	**R/L_Rud(−2), SB(70)**	**−10.5**	**17.3**
L_Flap(20), SB(70)	−23.0	13.3	**L_Flap(20), SB(70)**	**−7.2**	**16.0**
R/L_Rud(−4), SB(50)	**−8.1**	**11.3**	L_Flap(20), SB(60)	−19.0	23.1
R/L_Rud(2), SB(60)	−16.6	19.5	**L_Rud(−4), SB(70)**	**−6.3**	**14.8**
R/L_Rud(−2), SB(70)	−19.2	11.8	**SB(70), BF(−20)**	**−12.2**	**11.3**
SB(50), BF(−20)	−22.6	15.3	**SB(60), BF(−20)**	**−4.7**	**13.1**

[a]Trajectory reshaping disabled.

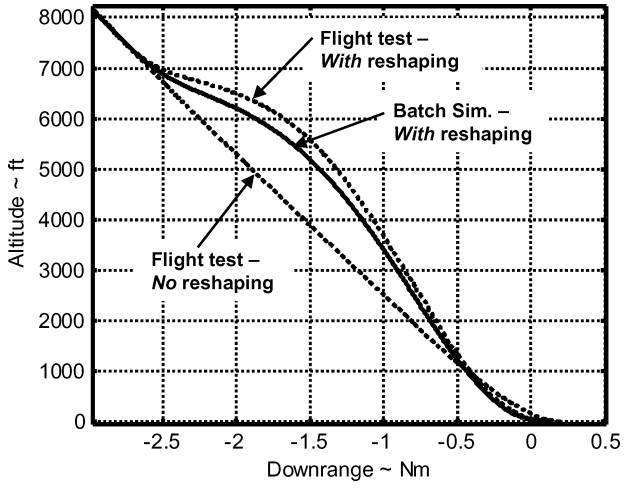

Fig. 13 Trajectory profile for example flight-test experiment.

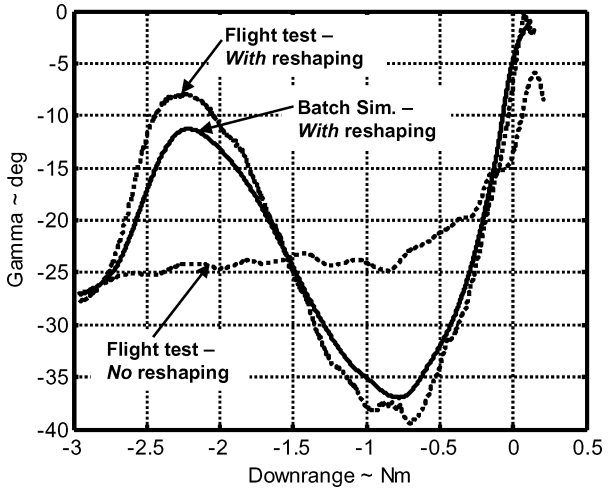

Fig. 14 Flight-path-angle histories for example flight-test experiment.

that the TIFS's direct-lift flaps saturated during the main flare and the model-following results were no longer valid.

Overall, however, the flight tests were quite successful, and under calm winds, several runs indicated results consistent with the batch-simulation experiments, validating the IAG&C designs. Last, note the two baseline runs (trajectory reshaping disabled) in the second and fourth table entries. Comparing these with their counterpart runs in which trajectory reshaping is enabled (the first and third table entries), it can be seen that trajectory reshaping improved the touchdown conditions.

Example results are presented below for the second pair of flight-test experiments (third and fourth table entries). This is a two-surface failure case requiring significant trajectory reshaping to achieve an acceptable touchdown. Here, the left ruddervator is locked in place at −4 deg and the speedbrake is locked hard over at 70 deg (up). Figure 13 presents the trajectory profiles, comparing the batch-simulation results with the flight-test results for the case with trajectory reshaping and the flight-test results for the case with no trajectory reshaping (baseline). The failure causes a substantial increase in drag, and the OPTG algorithm delivers a reshaped trajectory that maneuvers the vehicle to preserve energy in order to reach the runway. It can be seen that the vehicle first pulls up, reducing the glideslope to extend its range, and then dives again to increase the velocity in the main flare for added control-surface effectiveness. The differences seen between the flight test and batch simulation results are due to differences in the initial velocities. TIFS initial states could not be exactly matched with the corresponding batch-simulation runs, so there were often minor differences in the state histories. Note, however, that the TIFS eventually matches the altitude profile of the X-40A before the main flare. Figure 14 presents the flight-path-angle histories for this case. It can be seen that the touchdown flight-path angles for the cases with trajectory reshaping are approximately −2 deg, which is quite acceptable, leading to a soft touchdown. However, without trajectory reshaping, the flight-path angle at touchdown is approximately −8 deg, indicating a very hard landing. Again, the dramatic improvement in touchdown sink rate achieved by trajectory reshaping is evident by comparing the third and fourth entries in Table 6. Trajectory reshaping improved the touchdown sink rate from −21.9 to −7.2 fps.

Conclusions

Adaptation and reconfiguration technologies are of foremost interest in the development of next-generation launch platforms and will be essential for the success of future autonomous reusable launch vehicles. This paper presented a reconfigurable-control, adaptive-guidance, and trajectory-command-reshaping

system developed for the integrated adaptive guidance and control (IAG&C) program. This program has helped to prove out these advanced adaptive technologies and was in many ways a first step toward an implementable fault-tolerant landing system for autonomous reentry vehicles.

It was found that, in many cases, the reconfigurable controller is capable of recovering the vehicle following an effector failure. Control reconfiguration is achieved using an optimal allocation method. However, under severe or multiple control-effector failures, axis saturation and infeasibility of the nominal trajectory require guidance-loop adaptation and trajectory reshaping. Guidance-law adaptation is achieved by modeling the proportional gains in the guidance loops as functions of the controller's reference-model bandwidth. The trajectory-reshaping approach utilized for this program is known as the optimum-path-to-go (OPTG) algorithm. In the OPTG approach, a database of neighboring extremals is generated off line that covers all variations under consideration. It was determined that failure conditions can be adequately parameterized by upper and lower bounds on achievable lift and drag coefficients. This technique allows for rapid trajectory-database generation, avoiding the impracticable approach of modeling multitudes of possible failure scenarios. The OPTG methodology addresses the historical problems of on-board or real-time trajectory optimization. First, solutions are guaranteed to exist since they have already been determined off-line. Second, the computational burden is minimized through the numerical efficiency of using polynomial-based networks in flight to model the trajectory database.

The IAG&C technologies were flight tested using the total in-flight simulator (TIFS) research aircraft, simulating a modified version of Boeing's X-40A during the approach/landing flight phase. Flight-test experiments focused on locked-control-surface failure scenarios. For multiple locked-control-surface failures, it was determined that full recovery of nominal performance is often physically impossible with only control reconfiguration and guidance-gain adaptation. Under these situations, the OPTG trajectory-command-reshaping algorithm is typically needed to recover the mission and achieve a safe landing. Control-effector failures and subsequent control reconfiguration often bring about dramatic changes in the aerodynamic nature of the vehicle. Therefore, in-flight trajectory-command reshaping is frequently necessary to account for the changes in energy management brought about by the changes in the vehicle's lift and drag characteristics. For flight tests that were within X-40A specifications on acceptable winds, the majority of runs resulted in successful touchdown conditions, indicating the benefits of the IAG&C system. Furthermore, for these runs, the flight-test results were largely consistent with batch-simulation results, validating the IAG&C design.

Acknowledgments

Although the flight tests were supported under the IAG&C program, the development of the guidance law and the trajectory-reshaping algorithm was funded under two SBIR programs sponsored by 1) the Air Force Research Laboratory, David Doman, Technical Monitor, and 2) the Marshall Space Flight Center, G. Dukeman and John Hanson, Technical Monitors. This support is gratefully appreciated. Also, Marc Bodson at the University of Utah graciously provided the linear programming solver that was used in the control allocator. Acknowledgments are also due to Boeing, Huntington Beach, and General Dynamics Advanced Information Systems for their significant roles in the flight test program. Last, the original idea of interrogating (in flight) neural networks that represent optimal trajectory solutions is attributed to Roger Barron.

References

[1]Morring, F., "Orbital Space 'Plane' Could Be a Capsule," *Aviation Week and Space Technology*, 24 Feb. 2003, pp. 25, 26.

[2]Morring, F., "Heavy Lift," *Aviation Week and Space Technology*, 9 Feb. 2004, pp. 28, 29.

[3]"The Space Launch Initiative: Technology to Pioneer the Space Frontier," NASA Marshall Space Flight Center, Pub. 8-1250, FS-2001-06-122-MSFC, June 2001.

[4]Youssef, H., Chowdhry, R., Lee, H., Rodi, P., and Zimmerman, C., "Predictor–Corrector Entry Guidance for Reusable Launch Vehicles," AIAA Paper 2001-4043, Aug. 2001.

[5]Zimmerman, C., Dukeman, G., and Hanson, J., "An Automated Method to Compute Orbital Entry Re-Entry Trajectories with Heating Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, 2003, pp. 523–529.

[6]Dukeman, G., "Profile-Following Entry Guidance Using Linear Quadratic Regulator Theory," AIAA Paper 2002-4457, Aug. 2002.

[7]Barton, G., and Tragesser, S., "Autolanding Trajectory Design for the X-34," AIAA Paper 99-4161, Aug. 1999.

[8]Girerd, A., and Barton, G., "Next Generation Entry Guidance—Onboard Trajectory Generation for Unpowered Drop Tests," AIAA Paper 2000-3960, Aug. 2000.

[9]Mease, K., Chen, T., Teufel, P., and Schonenberger, H., "Reduced-Order Entry Trajectory Planning for Acceleration Guidance, *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 257–266.

[10]Leavitt, J., Saraf, A., Chen, D., and Mease, K., "Performance of Evolved Acceleration Guidance Logic for Entry (EAGLE)," AIAA Paper 2002-4456, Aug. 2002.

[11]Shen, Z., and Lu, P., "On-Board Generation of Three-Dimensional Constrained Entry Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, 2003, pp. 111–121.

[12]Ward, D., Monaco, J., and Schierman, J., "Reconfigurable Control for VTOL UAV Shipboard Landing," AIAA Paper 99-4045, Aug. 1999.

[13]Schierman, J., Ward, D., Monaco, J., and Hull, J., "A Reconfigurable Guidance Approach for Reusable Launch Vehicles," AIAA Paper 2001-4429, Aug. 2001.

[14]Schierman, J., Hull, J., and Ward, D., "Adaptive Guidance with Trajectory Reshaping for Reusable Launch Vehicles," AIAA Paper 2002-4458, Aug. 2002.

[15]Hanson, J., "Advanced Guidance and Control Project for Reusable Launch Vehicles," AIAA Paper 2000-3957, Aug. 2000.

[16]Hanson, J., "New Guidance for New Launchers," *Aerospace America*, Vol. 3, March 2003, pp. 36–41.

[17]Ward, D., Monaco, J., and Bodson, M., "Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, 1998, pp. 948–956.

[18]Bolender, M., and Doman, D., "Non-Linear Control Allocation Using Piecewise Linear Functions," AIAA Paper 2003-5357, Aug. 2003.

[19]Etkin, B., *Dynamics of Atmospheric Flight*, Wiley, New York, 1972, pp. 121–151.

[20]Doman, D., and Oppenheimer, M., "Improving Control Allocation Accuracy for Nonlinear Aircraft Dynamics," AIAA Paper 2002-4667, Aug. 2002.

[21]"Application of Multivariable Control Theory to Aircraft Control Laws," Wright Lab., Tech. Rept. WL-TR-96-3099, Wright–Patterson Air Force Base, OH, May 1996.

[22]Page, A., and Steinberg, M., "A Closed-Loop Comparison of Control Allocation Methods," AIAA Paper 2000-4538, Aug. 2000.

[23]Bodson, M., "Evaluation of Optimization Methods for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711.

[24]Schierman, J., Gandhi, N., Hull, J., and Ward, D., "Flight Test Results of an Adaptive Guidance System for Reusable Launch Vehicles," AIAA Paper 2004-4771, Aug. 2004.

[25]Sharma, M., "Flight-Path Angle Control via Neuro-Adaptive Backstepping," AIAA Paper 2002-4451, Aug. 2002.

[26]Powell, M., "Variable Metric Methods for Constrained Optimization," *Mathematical Programming: The State of the Art*, edited by A. Bachem, M. Grotschel, and B. Korte, Springer-Verlag, Berlin, 1983, pp. 288–311.

[27]Barron, R., Mucciardi, A., Cook, F., Craig, J., and Barron, A., *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, edited by S. J. Farlow, Dekker, New York, 1984, pp. 25–65.

[28]Ward, D., "Generalized Networks for Complex Function Modeling," *Proceedings, IEEE Systems, Man, and Cybernetics Conference (SMC-94)*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 1994.