

ON-LINE TRAJECTORY COMMAND RESHAPING FOR REUSABLE LAUNCH VEHICLES

J. D. Schierman,^{*} J. R. Hull,[†] and D. G. Ward[‡]
Barron Associates, Inc.
Charlottesville, Virginia

ABSTRACT

To enable autonomous operations in future Reusable Launch Vehicles (RLVs), onboard trajectory command reshaping will be required to facilitate recovery of the mission following a major anomalous event such as an effector failure. The Optimum-Path-To-Go (OPTG) on-line trajectory-reshaping algorithm is presented. In the OPTG methodology, a trajectory database is pre-computed *off-line* covering all variations under consideration. Then, polynomial-based networks are generated which map these variations to basis function coefficients that describe the shape of the trajectories. The networks are then interrogated *on-line*, and the resulting coefficients are used to generate trajectory commands. Thus, based on the current state of the system, the algorithm will reshape the commanded trajectory to give the best remaining path to the end of the mission segment. For this study, the commanded trajectory was reshaped on-line due to a severe multiple control surface failure. Without reshaping, the vehicle is lost, even with control reconfiguration and guidance adaptation. With trajectory reshaping, the mission is recovered.

INTRODUCTION

In recent years, NASA and the DOD have focused significant resources on developing advanced launch technologies. NASA's Space Launch Initiative (SLI) program was created to address many of the engineering goals as a part of this larger effort. The SLI program was recently restructured into the Orbital Space Plane (OSP) program and the Next Generation Launch Technology (NGLT) program [1]. These efforts will address manned launch/orbital vehicles, as well as completely autonomous space transportation systems. Of the original SLI program goals, an overriding concern was to develop launch platforms that are significantly safer, more reliable, and cheaper to develop/build [2,3]. *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 a minimal suite of control effectors. Because of this, failure of just one effector can often have devastating results 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 interface, or touchdown following the approach-to-landing flight phase) may be unachievable without altering the trajectory commands that drive the guidance loops. Such reshaped trajectories often result in reduced accelerations to account for the vehicle's degraded maneuvering capabilities. Or, the paths may be re-targeted to account for the vehicle's new lift and drag characteristics - while meeting certain critical constraints, such as heating constraints in hypersonic flight or pitch attitude at touchdown to avoid tail-scrape. Although control and guidance reconfiguration play essential roles in failure recovery, the focus of the paper will be on the final task, *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 results are presented and conclusions are then drawn from the results.

BACKGROUND

Prompted by programs such as SLI, onboard trajectory command generation approaches for 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 variations and navigation errors. However, these approaches do not address robustness to drastically changed vehicle dynamics because reconfiguration technologies were not yet available when these command generation

^{*} Senior Member, Senior Research Scientist

[†] Member, Research Scientist

[‡] Member, Senior Research Scientist

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 are usually quite computationally 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 (PC) approaches were presented in [4], [5] and [6]. In [4], an entry guidance design was presented with focus on thermal stress constraints. In [5] and [6], an approach for re-entry guidance was presented in which the trajectory solution is obtained prior to the de-orbit burn, then followed via an LQR guidance method during re-entry. Here too, heating constraints were the primary focus. The shooting method used in the procedure was aided by accounting for specific RLV re-entry trajectory constraints. An autoland guidance system was presented in [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 [9] and [10] for both the re-entry and TAEM flight phases. Another method for entry guidance was developed in [11]. Here too, the trajectory is computed prior to the de-orbit burn, and the solution lies in the middle of the "entry flight corridor" to assure 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 herein is known as the *Optimum-Path-To-Go* (OPTG) methodology, and has been successfully applied to numerous applications, including "smart" munitions [12], unmanned air vehicles [13] and several different RLV platforms [14,15]. This approach represents a unique solution to the onboard trajectory-reshaping problem. Unlike most other approaches, this method utilizes solutions pre-computed *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 on-line 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.

Two notable programs in this area of research are:

Marshall Space Flight Center's (MSFC) Advanced Guidance and Control (AG&C) program [16,17]: This project was established to examine several control and/or guidance methods to determine approaches that best address the goals of the SLI program. Although not part of the original effort, preliminary developments of the trajectory reshaping approach presented herein were supported through additional funds from the AG&C program [15].

The Air Force Research Lab's (AFRL) Integrated Adaptive Guidance and Control (IAG&C) program: The demonstration platform for this program is Boeing's X-40A RLV. This vehicle is an 80% scale version of Boeing's X-37 orbital RLV. The trajectory-reshaping algorithm presented in this paper was further matured for this program. This algorithm, along with an adaptive guidance system was integrated with a reconfigurable control system developed by AFRL.



Figure 1. Drop Test of Boeing's X-40A RLV.

In an earlier program, seven successful drop tests (from a carrier helicopter) of the X-40A were conducted, testing Boeing's original guidance and control systems for the approach-to-landing flight phase (see Figure 1). Because of this, the algorithms developed under the IAG&C program are being designed for this flight phase in the hopes of eventually repeating such drop tests. For risk reduction purposes, the algorithms will be flight tested using Veridian's Total In-Flight Simulator (TIFS) research aircraft, which will simulate the X-40A dynamics.

For the development presented herein, the control effectors modeled for the X-40A were:

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

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

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.

Bodyflap: this surface is hinged at the aft of the fuselage, similar in configuration to the Space Shuttle's body flap. Under nominal operations, this surface is used for pitch trim control, but may be used for active pitch control under failures.

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

SYSTEM ARCHITECTURE

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

Failure identification: subsequent to the failure, it is "recognized" either directly through sensor outputs, or indirectly through an onboard identification scheme. In previous developments, the OPTG algorithm worked in concert with the Modified Sequential Least Squares (MSLS) on-line identification technique [18].

Inner-loop control reconfiguration: the control system uses the 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. Significant research and development in the control reconfiguration problem has been accomplished over the past several years [18]. For the IAG&C program, AFRL's reconfigurable controller was developed in an optimal control allocator framework [19].

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. The IAG&C program utilizes a backstepping architecture for the guidance law [20]. Adaptation of the guidance law is triggered by control saturation, which is indicated by the inner-loop reconfigurable controller.

Onboard trajectory command reshaping: Given the crippled nature of the vehicle, 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 re-targeted trajectory that requires only forces and moments that can be achieved by the degraded capabilities of the vehicle while meeting certain critical constraints.

Figure 2 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. Examples of candidate critical parameters from past work include inner-closed-loop bandwidth for guidance law gain adaptation, and estimates of lift and drag coefficients and achievable bank angle for trajectory reshaping [14,15]. For the current study, the critical parameters for trajectory reshaping are *upper and lower bounds on achievable lift and drag forces*. These were found to be the most appropriate critical parameters for this study.

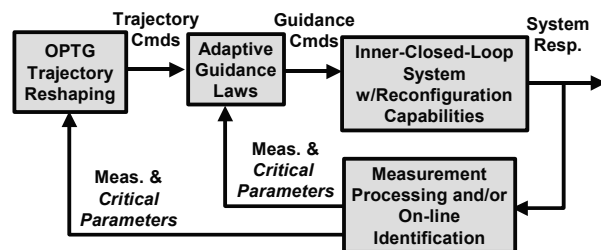


Figure 2. System Architecture with Trajectory Reshaping.

Note the small number of critical parameters required for the OPTG algorithm. A larger set of parameters is typically needed at the inner-loop control level, such as which surface has failed and in what position, or what elements have changed in the A and B matrices of the system's linear model. However, at the trajectory command level, 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 stuck control surface that would add approximately the same amount of additional axial force.

OPTG DESIGN PROTOCOL

The main steps in the OPTG design protocol are listed below and outlined in Figure 3.

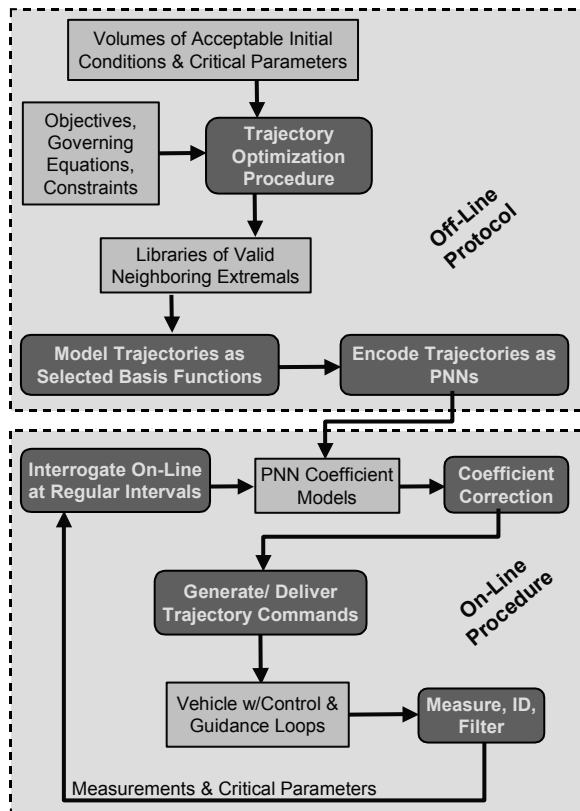


Figure 3. OPTG Framework.

Off-Line Trajectory Generation: 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. Once properly formulated, candidate trajectories can be readily obtained. A database of neighboring optimal extremals is generated by successively varying initial conditions and critical parameters. Such trajectories represent optimal flight paths for the vehicle under failure conditions. Libraries of trajectory databases are generated by repeating this procedure for several downrange locations (usually in equally spaced intervals). This large volume of trajectory data can be cumbersome to interrogate on-line with traditional table lookup methods. The next step in the OPTG approach solves this problem.

Off-Line Trajectory Encoding: This aspect of the OPTG methodology is the most important in terms of enabling on-line use. First, from the set of neighboring extremals found in Step 1, the states of the systems are modeled as a set of selected basis functions most appropriate for the mission segment. Polynomials in downrange have been selected here for the approach-to-landing flight phase, which is largely 2-dimensional flight (vertical plane motion). However, for example, the flight path is approximately helical in shape during the Heading Alignment Cone (HAC) within the Terminal Area Energy Management (TAEM) flight phase. Here, basis functions other than polynomials may be more appropriate.

A nonlinear function-modeling tool is then used to generate polynomial neural networks (PNNs) that map the current vehicle *observables* (states and critical parameters) to the coefficients of the basis functions describing the associated trajectory. These mappings relate the observables to the basis function coefficients in an efficient, compact manner.

On-Line Trajectory Integration: The final step in the OPTG approach is performed on-line. During flight, the current vehicle states and critical parameters are obtained from measurements or reconstructed from an on-line identification algorithm. This information is then used to compute an appropriate set of basis function coefficients from the on-board PNNs. 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 final conditions are met. The trajectory commands are then calculated from the corrected basis functions. The loop is closed by re-interrogating the functional mappings at regular intervals – typically on the order of 1 Hz. In this manner, trajectories are reshaped in flight to account for changes in the vehicle dynamics due to control surface failures or other significant anomalous events.

OPTG ADVANCEMENTS

The framework discussed above represents an improved design protocol for the OPTG approach. Previous versions of OPTG utilized the Calculus of Variations (COV) methodology to formulate the trajectory optimization problem [15]. The library of neighboring extremals was then encoded as a series of PNNs that mapped the states/critical parameters to the optimal *costates* (Lagrange multipliers) resulting from the constructed trajectory database. However, there were several problems with this approach.

Ill-conditioned PNNs: The PNNs were typically ill-conditioned immediately surrounding the nominal trajectory. Small changes in the states or critical parameters often resulted in large variations in the costates, and this behavior was difficult to model. It became necessary to partition the trajectory database around the nominal region and create several separate PNN models – greatly adding to the algorithm complexity. This behavior is not observed with the new basis function modeling implementation.

On-line correction: Another disadvantage of the former approach relates to on-line correction of the PNN costate estimates. Again, errors due to noise, disturbances, etc. require that the PNN estimates be corrected on-line to eliminate errors at the final boundary conditions. A Prediction and Correction of Costates (PCC) algorithm was developed for this purpose, and several simplifying assumptions were required for feasible real-time application. This was essentially a “stripped down” version of a shooting method that attempted to correct only the final position and ignored velocity and transversality conditions. Such an approach works well for munition or missile guidance applications. Unfortunately, for the RLV guidance problem, accurate estimates are required for all the states at touchdown, which does not allow for such simplifications. Running a similar PCC algorithm in real-time would pose a significant computational burden with no guarantee of convergence. However, as discussed later, the new method lends itself to a much simpler on-line PNN estimate correction algorithm.

COV method required: Most importantly, the prior approach relied on the use of the COV method in formulating the optimization problem because the encoded PNNs delivered costate information that was then used to propagate the governing equations. The COV approach, however, does not lend itself to rapid investigations of different objectives and more complicated constraints – often encountered when seeking the best trajectory optimization formulation. The new approach does not require the COV method, and we can therefore take advantage of numerical optimization procedures that easily allow for more complexity.

Next, the OPTG design for the IAG&C program will be discussed in detail.

OFF-LINE FORMULATION

System states appropriate for the approach-to-landing flight phase are:

- | | |
|-------------------|----------------------------------|
| (1) V = velocity | (2) γ = flight path angle |
| (3) X = downrange | (4) H = altitude |

with the corresponding governing equations of motion:

$$\begin{aligned}\dot{V} &= \frac{-D}{m} - g \sin(\gamma), \quad D = \bar{q} S C_D(\alpha, M) \\ \dot{\gamma} &= \frac{L}{mV} - \frac{g}{V} \cos(\gamma), \quad L = \bar{q} S C_L(\alpha, M) \\ \dot{X} &= V \cos(\gamma) \\ \dot{H} &= V \sin(\gamma)\end{aligned}\quad (1)$$

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, α = angle of attack, and M = Mach no.

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 re-entry guidance. An appropriate choice for approach-to-landing guidance is downrange position, X . To do this, each state equation is multiplied by

$$\frac{dt}{dX} = \frac{1}{V \cos(\gamma)} \quad (2)$$

resulting in:

$$\begin{aligned}\frac{dV}{dX} &= \dot{V} \frac{dt}{dX} = V_x = \frac{-D}{mV \cos(\gamma)} - \frac{g}{V} \tan(\gamma) \\ \frac{d\gamma}{dX} &= \dot{\gamma} \frac{dt}{dX} = \gamma_x = \frac{L}{mV^2 \cos(\gamma)} - \frac{g}{V^2} \\ \frac{dH}{dX} &= \dot{H} \frac{dt}{dX} = H_x = \tan(\gamma)\end{aligned}\quad (3)$$

Eq. (2) can be eliminated from consideration since 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.

Originally, it was thought that control surface saturation and degradations in achievable pitching moment would play major roles in the trajectory-reshaping problem. Therefore, early on, the optimization formulation included penalties on control deflection and pitch acceleration. However, anti-windup logic added to the reconfigurable control law solved the saturation problem. A penalty on pitch acceleration was investigated to minimize required pitching moment, but this necessitated inclusion of the pitching moment equation of motion in the formulation. Yet, the fast pitch dynamics coupled with the slower translational dynamics made the problem too stiff, and optimal solutions were difficult to obtain. This was overcome by indirectly minimizing required pitching moment

through a penalty term on flight path angle rate. Yet, this penalty term was not used in the final formulation because pitching moment limits were never seen to be violated – even for the most severe failure cases, and solutions were much easier to obtain without this term. Other vehicles or other flight regimes, though, may necessitate inclusion of a penalty related to pitching moment.

However, it was determined that the trajectory commands should be reshaped when failures result in a degraded ability to generate *lift and drag forces* required to follow the nominal trajectory. This was typically the case for more severe, multiple surface failure cases. Therefore, the goal of reshaping the commanded trajectory in the event of a failure was to provide admissible trajectories resulting in lift and drag profiles that were well within vehicle’s achievable limits.

The optimization problem was formulated as a Successive Quadratic Programming (SQP) [21] problem. For the SQP method, the states and command or decision variables are partitioned into N discrete points. At each point, an approximation is made that generates a solvable quadratic programming *subproblem* whose solution is used to form the search direction. Optimization results were investigated for several values of N , and it was found that $N = 70$ gives the best results for this particular problem.

Because failures significantly change the aerodynamic characteristics of the vehicle, it was found that different optimization formulations were required under nominal and failure conditions. For the nominal system, angle-of-attack was used as the as the decision variable, and the optimization problem was formalized as follows:

$$\min_{\alpha \in \mathbb{R}^N} J_{nom} = \alpha^T \alpha \quad (4)$$

subject to:

$$\alpha_{\min_trim} \leq \alpha_i \leq \alpha_{\max_trim}, i = 1, \dots, N \quad (5)$$

and the following terminal constraints:

$$\begin{aligned} X_{f_{\min}} \leq X_f \leq X_{f_{\max}}, \quad V_f = V_{f_{\max}} \\ \dot{H}_f \geq \dot{H}_{f_{\min}}, \quad H_f = H_{Rmvy} \end{aligned} \quad (6)$$

The objective of the nominal cost function is to minimize angle-of-attack. This is done so that the vehicle has margin in angle-of-attack near touchdown to further arrest the sinkrate if needed, and also to minimize the pitch attitude at touchdown so that tail scrape constraints are not violated (pitch attitude, $\theta < 15$ deg.) .

The physical constraint on angle-of-attack in Eq. (5) keeps the vehicle within its trimmable region. For the nominal vehicle, $-5^\circ \leq \alpha_{trim} \leq +15^\circ$. (For the failure

cases, these bounds can significantly reduce.) Finally, Eq. (6) gives the terminal constraints. 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 was defined to be: $0 \text{ ft} \leq X \leq 6000 \text{ ft}$. For the final velocity, $V_{f_{\max}} = 300 \text{ fps}$. Much beyond this value and tire blowout can occur. Nevertheless, this velocity is desired at touchdown to maximize authority in the control surfaces. For touchdown sink rate, $\dot{H}_{f_{\min}} = -3 \text{ fps}$. The actual sink rate limit is approximately -10 fps , before damage to the vehicle may occur. However, margin is built into the optimization as the touchdown sink rate can increase with real-world effects/disturbances. Lastly, the final altitude is set to the runway altitude, H_{Rmvy} .

Note that the state histories are found by numerically integrating the system in Eq. (3) from X_o to X_f with the N decision variables in the command vector α . In this manner, the state equations are satisfied even though they are not directly defined as constraints.

For failure cases, coefficient of lift, C_L , was used as the decision variable, and the optimization problem was formalized as follows:

$$\min_{C_L \in \mathbb{R}^N} J_{fail} = -k_1 V_{f_{\max}} + k_2 (C_L - C_{L_{nom}})^T (C_L - C_{L_{nom}}) \quad (7)$$

subject to:

$$C_{L_{\min_trim}} \leq C_{L_i} \leq C_{L_{\max_trim}}, i = 1, \dots, N \quad (8)$$

and the following terminal constraints:

$$\begin{aligned} X_{f_{\min}} \leq X_f \leq X_{f_{\max}}, \quad V_f \leq V_{f_{\max}} \\ \dot{H}_f \geq \dot{H}_{f_{\min}}, \quad H_f = H_{Rmvy} \end{aligned} \quad (9)$$

The objective of the cost function for failure cases is to maximize final velocity and keep the lift coefficient close to the nominal value (k_1, k_2 are tuning parameters). Final velocity is maximized to give as much authority to the control surfaces as possible near touchdown when it is critical that the vehicle be able to maneuver to correct errors in tracking the commanded trajectory. Finally, note that a penalty term on angle-of-attack was not included in the cost function for the failure cases studied because the vehicle typically required at or near maximum allowed angles-of-attack to recover the mission.

The constraint of Eq. (8) was used to model variations in upper/lower bounds on lift. Markedly different trajectory solutions were obtained for variations in these bounds. Drag was modeled as a linear function of lift (discussed more in the next section). Due to errors, approximations and disturbances, the actual vehicle will almost certainly fly lift and drag profiles that differ

somewhat from the optimal solution. However, the vehicle will follow the commanded trajectory as long as it has the capability to achieve the required lift and drag forces. It is for this reason that upper/lower bounds on lift and drag were chosen as the critical parameters.

Lastly, the terminal constraints given in Eq. (9) are the same as for the nominal case except for the *inequality* constraint on final velocity. This was set as an equality constraint for the nominal system, and was easily achieved. Hence, a penalty involving final velocity in the cost function for the nominal system was not needed. For failure cases, stuck surfaces typically add drag, and it was usually the case that touchdown velocities were much less than $V_{f_{max}}$.

OFF-LINE TRAJECTORY DATABASE GENERATION/ENCODING

The next step is to generate databases of neighboring solutions to the optimization problem. For the IAG&C program, studies were limited to variations in states/critical parameters at the beginning of the approach-landing segment. Therefore, only one trajectory database was generated for this study. (In an earlier study involving the X-33 RLV, databases were generated at several downrange locations for variations in drag due to a speedbrake failure [15]).

In developing the trajectory database, the aerodynamic characteristics for a significant number of failure cases were first investigated. This was done to develop an appropriate data set of critical parameters as well as appropriate lower order aerodynamic models. Figure 4 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/bodyflap failure. Note that as the severity of the failure increases, the range between minimum to maximum C_D and C_L reduces significantly. Also shown in the background of Figure 4 are the minimum and maximum $\{C_L, C_D\}$ points for the actual failure cases generated. This data set was used in modeling uniform distributions for max/min C_L and C_D values that formed the set of admissible critical parameters. Then, for each trajectory generated, C_D was simply modeled as a linear function of C_L between the chosen min/max $\{C_L, C_D\}$ end points. This approximation has not been seen to degrade the performance of the OPTG trajectory reshaping when integrated with the high-fidelity 6-dof-simulation model.

This approach provides a method whereby a trajectory database can be rapidly generated by *intelligently* choosing the critical parameters, coupled with an appropriate linear relationship between lift and drag. The database was *not* generated by introducing

different failures for each trajectory and using C_L and C_D models specific to that particular failure. Given all possible combinations of failed surfaces and failed deflection values, such a protocol would be too cumbersome to be a practical approach.

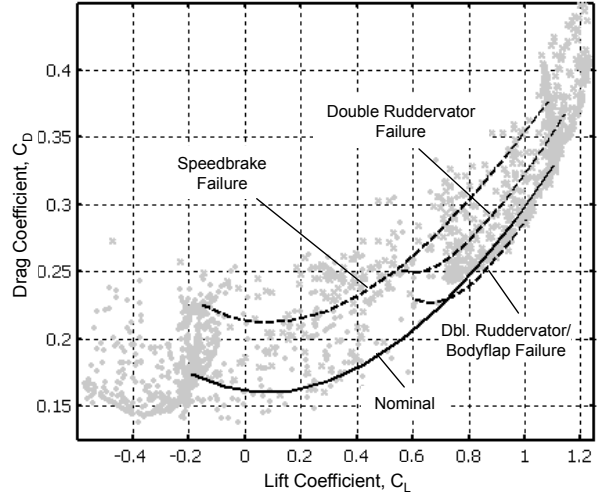


Figure 4. Aerodynamic Characteristics Under Failures.

Once the database is generated, the velocity and altitude states are then modeled as polynomials in downrange, X , via a least squares curve fitting routine. That is:

$$\begin{aligned} V_{cmd} &= a_n X^n + \dots + a_2 X^2 + a_1 X + a_0 \\ H_{cmd} &= b_n X^n + \dots + b_2 X^2 + b_1 X + b_0 \end{aligned} \quad (10)$$

Best fits were found when these polynomials are 5th order for the nominal case, 6th order for the failure cases studied. Then, for each trajectory, the polynomial coefficients are mapped to the initial states and upper/lower bounds on C_D and C_L corresponding to that trajectory. Again, these mappings are polynomial neural networks (PNNs) and are automatically generated through the use of the software package *GNOSIS* [22].

ON-LINE TRAJECTORY RESHAPING

Estimates of the critical parameters are determined on-line using a Piecewise Linear Programming (PLP) algorithm [23]. Given the failed control surface positions, this algorithm solves a linear programming problem on-line to give trim lift and drag models. These models then provide the upper/lower bounds on lift and drag.

The PNN models that relate the current state measurements and the critical parameter estimates are then interrogated, resulting in *uncorrected* polynomial coefficients for the velocity and altitude commands. Recall, a coefficient correction algorithm is then used to correct for errors and disturbances not modeled in the PNNs. Here, a constrained least-squares algorithm is employed where the objectives are:

1. correct the coefficients to eliminate errors in the initial 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 then,

$$J = \frac{1}{2} \left\| \sqrt{W_1} (Y - \tilde{\theta}^T \Phi) \right\|_2 + \frac{1}{2} \left\| \sqrt{W_2} (\tilde{\theta} - \theta_o) \right\|_2 \quad (11)$$

where Y is the polynomial output at the initial and final boundaries, Φ is the regressor at the initial and final boundaries, $\tilde{\theta}$ is the corrected coefficients, and θ_o is the uncorrected coefficients given by the PNNs. W_1 and W_2 are relative weighting matrices used for tuning the solution. The solution is given by:

$$\tilde{\theta} = \left[\Phi_f^T W_1 \Phi_f + W_2 + R \right]^{-1} \left[\Phi_f^T W_1 Y_f + W_2 \theta_o \right] \quad (12)$$

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

$$Y = \begin{bmatrix} V_i \\ V_f \end{bmatrix}, \quad \Phi = \begin{bmatrix} X_i^n, \dots, X_i^2, X_i, 1 \\ X_f^n, \dots, X_f^2, X_f, 1 \end{bmatrix}^T \quad (13)$$

However, for altitude, it is also desired to match the slope at the initial and final boundaries (slope = $\partial H / \partial X = \tan(\gamma)$ - see Eq. (3)). Therefore, in this case,

$$Y = \begin{bmatrix} H_i \\ \tan(\gamma_i) \\ H_f \\ \tan(\gamma_f) \end{bmatrix}, \quad \Phi = \begin{bmatrix} X_i^n, \dots, X_i^2, X_i, 1 \\ nX_i^{n-1}, \dots, 2X_i, 1 \\ X_f^n, \dots, X_f^2, X_f, 1 \\ nX_f^{n-1}, \dots, 2X_f, 1 \end{bmatrix}^T \quad (14)$$

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

$$\begin{aligned} V_{cmd}^* &= a_n^* X^n + \dots + a_2^* X^2 + a_1^* X + a_o^* \\ H_{cmd}^* &= b_n^* X^n + \dots + b_2^* X^2 + b_1^* X + b_o^* \end{aligned} \quad (15)$$

Flight path angle and altitude rate commands are subsequently derived from the equations of motion:

$$\begin{aligned} \gamma_{cmd}^* &= \tan^{-1}(\partial H_{cmd}^* / \partial X) \\ \dot{H}_{cmd}^* &= V_{cmd}^* \sin(\gamma_{cmd}^*) \end{aligned} \quad (16)$$

These commands are then used to drive the guidance loops.

EXPERIMENTAL RESULTS

Flight of the X-40A during approach-to-landing was studied under both nominal conditions (all control surfaces working) and for a significant number of control effector failure cases. For this investigation, the type of failure was limited to control surfaces that were

stuck or jammed in-place. Table 1 presents a summary of failure *classes* studied. Each failure class 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, for single ruddervator failures, cases were studied in 1 deg. increments between ± 5 deg., and in 5 deg. increments outside that range, ± 10 deg. to ± 30 deg. (for a total of 21 cases). The other failure classes were set up in a similar fashion. Larger numbers of cases were studied for multiple surface failures.

Table 1. Summary of Failure Classes Investigated.

Failure Case	Level of Required Reconfiguration/Adaptation		
	Inner-Loop Control	Outer-Loop Guidance	Trajectory Reshaping
Bodyflap	✓	✗	✗
Speedbrake	✗	✗	✓*
Single ruddervator	✓	✓**	✓**
Single flaperon	✓	✗	✗
Speedbrake/Bodyflap	✓	✗	✗
Double ruddervator	✓	✓**	✓**
Double flaperon	Unrecoverable		
Bodyflap/single ruddervator	✓	✓	✓
Bodyflap/double ruddervator	✓	✓	✓

* Reshaping not required, but can provide benefit.

** Only required for larger stuck deflection angles.

More importantly, Table 1 indicates the level of reconfiguration or adaptation required for the different classes of failures. For single surface failures, recovery is possible with inner-loop control reconfiguration alone, except for large ruddervator stuck deflections, which required guidance adaptation and trajectory reshaping. For most speedbrake failures, the baseline system (with no reconfiguration/adaptation capabilities) can typically recover the mission. However, severe speedbrake failures (fixed at > 50 deg.) coupled with errors and disturbances such as wind gusts may require some adaptation. In this case, trajectory reshaping may give benefit in providing more robustness with an “easier” trajectory to follow. Next, note that speedbrake/bodyflap failures can also be recovered with only control reconfiguration. This is not an unexpected result since the original X-40A vehicle does not possess these effectors. Conversely, for the majority of double ruddervator failures, recovery is possible with control reconfiguration alone primarily because the speedbrake and bodyflap take over the pitch control function. Only very large stuck deflections require trajectory reshaping. The double flaperon failure is unrecoverable chiefly because there is no way of physically regaining roll control. Finally, the shaded rows in Table 1 indicate the class of failures that are of primary interest. It is seen that combination bodyflap/single or double ruddervator failures require all three levels of

reconfiguration (control, guidance and trajectory reshaping) in order to recover the mission. This is true even for small stuck deflection angles. Experimental results presented below will focus on a bodyflap/double ruddervator failure case.

The following figures present simulation results using the high fidelity 6-dof 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. For this case study, the bodyflap is failed at -20 deg. (trailing edge down) and both ruddervators are failed asymmetrically (do not cause roll/yaw) at +2 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.

Figure 5 presents the altitude (in feet) plotted versus downrange (in nautical miles). Although not shown, with no failures the vehicle is able to follow the nominal commanded trajectory quite accurately. The nominal commanded trajectory was obtained from the OPTG algorithm and is similar to Shuttle-heritage approach-to-landing profiles. The longest portion of such trajectories is made up of a steep glideslope, followed by a main flare to a short, shallow glideslope. A final flare at the end of the trajectory arrests the sink rate just before touchdown.

It can be seen that the trajectory under the failure condition, but *without* command reshaping, cannot follow the nominal commanded trajectory. The touchdown sink rate for this case is nearly -30 fps. If not completely destroyed, the vehicle would most certainly suffer severe damage. In contrast, the trajectory under the failure condition *with* command reshaping can accurately follow the *reshaped* commanded trajectory. Here, the touchdown sink rate is -4 fps, well within acceptable limits.

Figure 6 presents the flight path angle histories. 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. Touchdown flight path angles greater than -2 deg. are considered acceptable. The touchdown flight path for no reshaping is approximately -7 deg., whereas with reshaping it is approximately -1 deg.

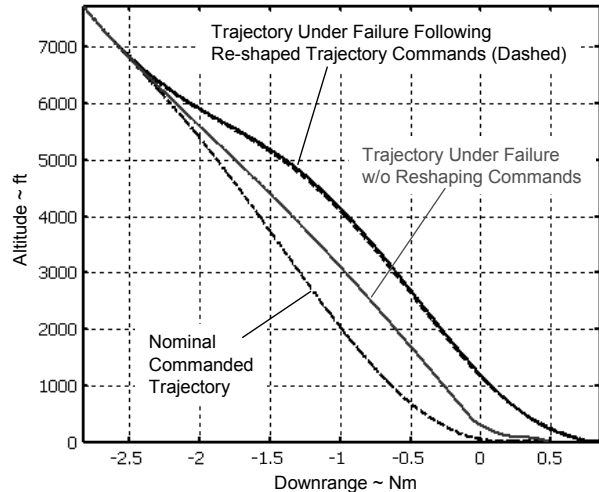


Figure 5. Trajectory Profile.

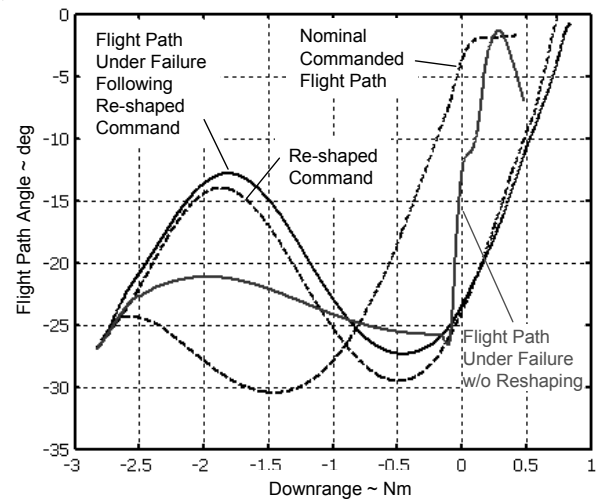


Figure 6. Flight Path Angle Histories.

The angle-of-attack histories are shown in Figure 7. It can be seen that for the case with command reshaping, the angle-of-attack 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.

Figure 8 presents the lift coefficients for the nominal and failure conditions during the steep glideslope portion of the flight, where the vehicle is most likely to be in pitch trim. The upper/lower bounds on trim C_L are also shown for the nominal and failure cases. These are considered quite accurate estimates of the true trim bounds, and were generated solving a nonlinear programming problem involving the high-fidelity aerodynamic model. It can be seen that the failure causes these bounds to close in, severely reducing the trimmable range. The minimum bound on lift notably

increases - due largely to the bodyflap stuck in a trailing edge down position.

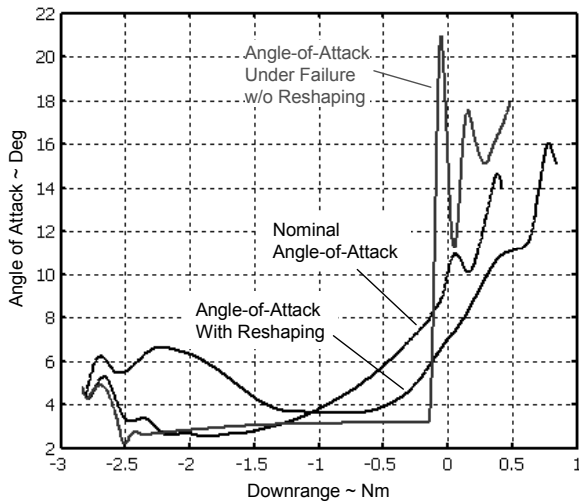


Figure 7. Angle-of-Attack Histories.

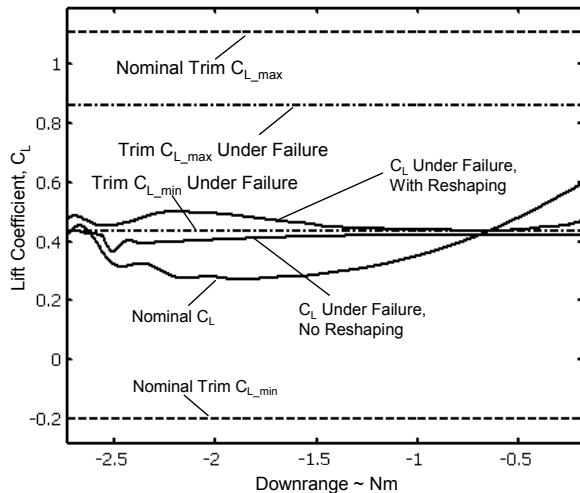


Figure 8. Lift Coefficient and Max/Min Bounds.

Figure 8 indicates that the failed vehicle without command reshaping is not in trim, and attempts to fly at lower C_L values than the minimum bound on trim C_L . However, it cannot lower the C_L enough to be close to the nominal C_L , hence, it cannot track the nominal trajectory. In contrast, the C_L for the vehicle with command reshaping lies within the bounds for the failure case. This indicates that the critical parameters were accurately estimated on-line, and the OPTG algorithm correctly supplied a reshaped trajectory that would not require the lift to violate these bounds. For this reason, the crippled vehicle is able to follow the reshaped trajectory.

Note that for the case with command reshaping, the pitch attitude at touchdown is approximately 14 deg (nominal is around 13 deg.). This is near the allowable limit, and with little margin, tail scrape could possibly

occur. However, this is clearly better than the alternative - severe structural damage to the vehicle. Note that maximum touchdown velocity was not violated for any of the cases. In fact, higher velocities than those obtained are desired. The nature of the failure adds drag to the vehicle, causing it to slow down. Lastly, although not shown, deflection histories for the remaining working surfaces were seen to be reasonable for the reshaped trajectory. The flaperons were near saturation for only a brief period of time during the steep glideslope.

SUMMARY AND CONCLUSIONS

Reconfiguration capabilities will be essential for the success of future autonomous Reusable Launch Vehicles (RLVs). RLVs typically have a minimal set of control effectors, and full recovery of nominal maneuvering capabilities may be physically impossible for a large set of failure scenarios. Because of this, there will be three main elements to RLV reconfiguration: control reconfiguration, guidance adaptation and trajectory reshaping. Final conditions of the mission segment may be unachievable if the trajectory commands driving the guidance loops are not reshaped on-line.

The focus of the paper was on the Optimum-Path-To-Go (OPTG) trajectory-reshaping algorithm. As presented, the algorithm reflects several advancements, and gives much improved results over previous versions with a greatly simplified formulation. The OPTG methodology consists of three main steps. First, a trajectory optimization problem is formulated as a tool to generate a database of neighboring extremals *off-line* that covers all variations under consideration. Failure conditions are parameterized by upper and lower bounds on achievable lift and drag coefficients. This approach allows for rapid trajectory database generation, avoiding the infeasibility of modeling multitudes of possible failure scenarios. Next, the database is efficiently encoded using polynomial-based networks, which map vehicle states and lift and drag bounds to basis function coefficients that model the trajectories. The networks are then interrogated *on-line* to generate the trajectory commands that drive the guidance loops.

Extensive failure analyses and simulation experiments were performed using a modified version of Boeing's X-40A RLV. The focus of the failure experiments was on the approach-to-landing flight phase. Simulation results for a bodyflap/double ruddervator failure were presented. Here, three of the vehicle's six control surfaces were failed, and this represents the one of the most severe failure cases from which recovery is possible. Simulation results showed that even with control reconfiguration and guidance adaptation, the

mission could not be recovered without additionally reshaping the commanded trajectory.

ACKNOWLEDGMENTS

This work was funded under two SBIR programs sponsored by (1) the Air Force Research Laboratory, Dr. David Doman, Technical Monitor, and (2) the Marshall Space Flight Center, Mr. Greg Dukeman and Dr. John Hanson, Technical Monitors. *Their support is gratefully appreciated.* Acknowledgments are also due to Boeing, Huntington Beach, which aided in the development of the demonstration model, and to Ms. Neha Gandhi for her help in providing experimental results.

REFERENCES

1. Morring, F., "Orbital Space 'Plane' Could Be a Capsule," *Aviation Week and Space Technology*, Feb. 24, 2003, pp. 25-26.
2. Anon., "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.
3. Anon., "Introduction to NASA's Integrated Space Transportation Plan and Space Launch Initiative," NASA White Paper, May 17, 2001.
4. Youssef, H., Chowdhry, R., Lee, H., Rodi, P., Zimmerman, C., "Predictor-Corrector Entry Guidance for Reusable Launch Vehicles," AIAA-2001-4043, Proc. AIAA Guidance, Navigation, and Control Conf., Montreal, Canada, August, 2001.
5. Zimmerman, C., Dukeman, G., Hanson, J., "An Automated Method to Compute Orbital Re-entry Trajectories with Heating Constraints," AIAA-2002-4454, Proc. AIAA Guidance, Navigation, and Control Conf., Monterey, CA, August, 2002.
6. Dukeman, G., "Profile-Following Entry Guidance Using Linear Quadratic Regulator Theory," AIAA-2002-4457, Proc. AIAA Guidance, Navigation, and Control Conf., Monterey, CA, August, 2002.
7. Barton, G., Tragesser, S., "Autolanding Trajectory Design for the X-34," AIAA-99-4161, Proc. AIAA Guidance, Navigation, and Control Conf., Portland, OR, August, 1999.
8. Girerd, A., Barton, G., "Next Generation Entry Guidance – Onboard Trajectory Generation for Unpowered Drop Tests," AIAA-2000-3960, Proc. AIAA Guidance, Navigation, and Control Conf., Denver, CO, August, 2000.
9. Mease, K., Chen, T., Teufel, P., Schonenberger, H., "Reduced-Order Entry Trajectory Planning for Acceleration Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, March-April 2002, pp. 257-266.
10. Leavitt, J., Saraf, A., Chen, D., Mease, K., "Performance of Evolved Acceleration Guidance Logic for Entry (EAGLE)," AIAA-2002-4456, Proc. AIAA Guidance, Navigation, and Control Conf., Monterey, CA, August, 2002.
11. Shen, Z., Lu, P., "On-board Generation of Three-Dimensional Constrained Entry Trajectories," AIAA-2002-4455, Proc. AIAA Guidance, Navigation, and Control Conf., Monterey, CA, August, 2002.
12. Bateman, A., Hull, J., Ward, D., Monaco, J., "Hydra-7 Guidance Law Development: Year 2," Barron Associates, Inc. Final Technical Report, Submitted to Lockheed-Martin Missiles and Fire Control-Advanced Projects, May 17, 2001.
13. Ward, D., J. Monaco, and J. Schierman, "Reconfigurable Control for VTOL UAV Shipboard Landing," AIAA-99-4045, Proc. AIAA Guidance, Navigation, and Control Conf., Portland, OR, Aug. 1999.
14. Schierman, J., Ward, D., Monaco, J., Hull, J., "A Reconfigurable Guidance Approach for Reusable Launch Vehicles," AIAA-2001-4429, Proc. AIAA Guidance, Navigation, and Control Conf., Montreal, Canada, August, 2001.
15. Schierman, J., Hull, J., Ward, D., "Adaptive Guidance with Trajectory Reshaping for Reusable Launch Vehicles," AIAA-2002-4458, Proc. AIAA Guidance, Navigation, and Control Conf., Monterey, CA, Aug. 2002.
16. Hanson, J. M., "Advanced Guidance and Control Project for Reusable Launch Vehicles," AIAA-2000-3957, Proc. AIAA Guidance, Navigation, and Control Conf., Denver, CO, August, 2000.
17. Hanson, J., "New Guidance for New Launchers," *Aerospace America*, March, 2003, pp. 36-41.
18. Ward, D., J. Monaco, and M. Bodson, "Development and flight testing of a parameter identification algorithm for reconfigurable control," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, Nov. – Dec. 1998, pp. 948-956.
19. Doman, D., Oppenheimer, M., "Improving Control Allocation Accuracy for Nonlinear Aircraft Dynamics," AIAA-2002-4667, Proc. AIAA Guidance, Navigation, and Control Conf., Monterey, CA, Aug. 2002.
20. Sharma, M., "Flight-Path Angle Control via Neuro-Adaptive Backstepping," AIAA-2002-4451, Proc. AIAA Guidance, Navigation, and Control Conf., Monterey, CA, Aug. 2002.
21. Fletcher, R., *Practical Methods of Optimization, Vol. 2, Constrained Optimization*, John Wiley and Sons, 1980.
22. Ward, D., "Generalized Networks for Complex Function Modeling," Proc. IEEE Systems, Man, and Cybernetics (SMC-94) Conf., Oct. 2-5, 1994.
23. Bolender, M. A. and Doman, D.B., "Non-linear Control Allocation Using Piecewise Linear Functions," AIAA-2003-5357, Proc. AIAA Guidance, Navigation, and Control Conf., Austin, TX, Aug. 2003.