

ADAPTIVE GUIDANCE WITH TRAJECTORY RESHAPING FOR REUSABLE LAUNCH VEHICLES

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

Abstract

Reusable launch vehicle (RLV) designs are influenced by weight and other constraints that rarely allow for significant effector redundancy. Therefore, reconfiguration of the inner-loop control system and outer-loop guidance functions can be necessary to compensate for significant control effector failures, aerodynamic uncertainties, and disturbances such as strong winds. Trajectory reshaping is also important in some situations to judiciously manage vehicle energy on-line to meet the flight objectives. This paper presents an Optimum-Path-To-Go (OPTG) algorithm, which is a general framework to perform the on-line trajectory-command generation task. The methodology was applied to Lockheed's X-33 RLV for the approach-and-landing phase of flight, and a Monte-Carlo simulation analysis was used to demonstrate the benefits of the approach. Random increments to the vehicle drag were inserted in the simulation at various downrange locations. Such variations are representative of mismodeled aerodynamics, speedbrake failure, or significant winds. The X-33 implementation of the OPTG was able to reshape the trajectories to result in a safe landing for greater than 93% of the cases. Without the OPTG capability, safe landing was accomplished for only 29% of the drag variations simulated.

Introduction

A central focus for NASA's *Space Launch Initiative (SLI) Program* is to develop a new launch system architecture that is significantly safer, more reliable and less costly [1,2]. SLI developed technology will form the foundation of NASA's long-range Integrated Space Transportation Plan, which will eventually replace the aging fleet of Space Shuttles. Full-scale development of a 2nd generation Reusable Launch Vehicle (RLV)

system is planned for the middle of this decade, becoming operational by early next decade.

To address the goal of increasing vehicle safety and reliability, there is now significant interest in *reconfiguration* technologies for guidance, control and avionics systems. The capability for on-line reconfiguration of the control and guidance systems is now considered an enabling technology for next generation launch vehicles [3,4,5]. Some of the reasons for this are as follows:

1. The amount of aerodynamic uncertainties for RLVs is significantly greater than that of other flight vehicles due to the difficulty in obtaining reliable wind-tunnel data for the high altitude, hypersonic flight conditions at which RLVs must operate. If fixed-structure controllers are used, based on largely erroneous models, then degraded performance or instability may result. Here, adaptation and reconfiguration can provide an important benefit.
2. Weight restrictions are critical for launch vehicles due to the high cost of placing payloads into orbit. Because of this, typically there will not be as much actuation redundancy or alternate control effectors in an RLV as there are in commercial or military aircraft. Large potential savings can be realized if the hardware redundancy function can be replaced with software (i.e. reconfigurable control system algorithms) such that the probability of mission failure is at least equivalent, if not improved.

Over the past decade, a number of researchers have developed reconfigurable *control* systems for a variety of flight vehicles with promising results (e.g. [6,7,8,9]). The majority of these techniques have been demonstrated to recover nominal or near-nominal *inner-closed-loop* performance for systems with sufficient control effector redundancy. Under these circumstances, effector failure and subsequent *control* reconfiguration is *transparent* to the pilot or outer-loop autonomous guidance system, which is able to continue nominal operations. Therefore, there has not been a critical need for *adaptive outer-loop guidance laws*. That scenario, however, is no longer applicable for RLV systems. Although inner-loop control reconfiguration will still be essential to recover attitude

^{*} Senior Research Scientist; Senior Member, AIAA

[†] Research Scientist; Member, AIAA

[‡] Senior Research Scientist; Member, AIAA

stability and provide a measure of axis decoupling, for a large set of failure scenarios, full recovery of nominal flying qualities may be physically unrealizable due to their minimal suite of control effectors. In these cases, the guidance system must be capable of reconfiguring to adapt to a vehicle with reduced maneuvering capabilities.

These challenges have led to the desire to investigate not only reconfigurable control systems for RLVs, but also reconfigurable guidance systems as well. The work presented herein was funded by Marshall Space Flight Center's (MSFC) Advanced Guidance and Control (AG&C) project, which was established to examine several alternative reconfigurable control and/or guidance methods to see which ones best address the needs of RLVs and the goals of the SLI program [10]. The focus of the work presented here is on the reconfigurable guidance problem, and Lockheed's X-33 RLV served as the demonstration platform.

The next section discusses the general reconfigurable guidance methodology. Following this, the main body of the paper involves the trajectory-reshaping algorithm. Experimental results are then presented. Lastly, a summary of the work is presented and conclusions are given.

Guidance Methodology

The on-line guidance reconfiguration approach presented here has been developed and matured over the course of several years [5]. Figure 1 illustrates the general feedback architecture of this approach. This approach is novel in its modularity. Our design goal is to enable the guidance system to be integrated with any valid inner-loop reconfigurable control system. Such an approach has distinct advantages when different organizations design the inner-loop control and outer-loop guidance systems, or reconfiguration is augmented in a retrofit manner to an existing control/guidance system.

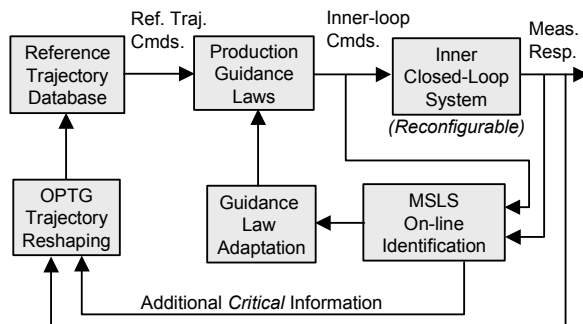


Figure 1. Reconfigurable Guidance Feedback Architecture.

The main aspects of the overall feedback system are:

Control Reconfiguration Task - The inner-closed-loop reconfigurable control system is tasked to maintain *attitude* stability and recover maneuvering performance (such as axis-decoupling) to the extent possible in the face of, for example, a catastrophic effector failure. Significant research and development in the control reconfiguration problem has been accomplished over the past several years [6-9]. As indicated in Figure 1, we assume here that the inner-closed-loop system has control reconfiguration capabilities. The current development does not explicitly model such a system in detail, but rather models the net effects of control reconfiguration that are important to the guidance task.

Guidance Reconfiguration Task - The reconfigurable guidance system is tasked to follow reference trajectory commands and recover a measure of *flight* stability margins in the face of degraded vehicle performance. More information regarding our guidance reconfiguration approach can be found in [5]. A guidance gain reconfiguration algorithm was developed under this program and achieved excellent recovery performance. However, again, the focus of this paper is on the trajectory-reshaping task, discussed next.

Trajectory Reshaping Task - The final task in the guidance system involves reshaping trajectory commands. Here, the Optimum-Path-To-Go (OPTG) algorithm is employed [5]. Trajectory reshaping may often be required to achieve the desired end conditions of the mission segment - even with both inner-loop control and outer-loop guidance reconfiguration.

Again, a main goal in our approach is to develop a modular architecture, allowing for independent designs of the control and guidance systems. To achieve this modularity, an identification algorithm known as Modified Sequential Least Squares (MSLS) [8] is employed in the outer-loops (see Figure 1). In the design and development of the reconfigurable guidance system, the inner-closed-loop is represented as a Low-Order-Equivalent-System (LOES) that captures the dynamic characteristics important to the guidance task (e.g. modeling vehicle attitude dynamics is typically not required at this stage, especially since the control loops are closed). Because of this, the number of parameters to identify or estimate, and the number of system dynamic changes required to be addressed are significantly reduced compared to the inner-loop control reconfiguration task. We believe that the multitude of dynamic changes from which recovery is possible may be characterized by a small set of key *critical parameters* at the guidance level, and the "curse of dimensionality" will not be an issue. (It is these critical parameters that are identified by the MSLS algorithm.) As an example, consider the case of an

elevon failure for an RLV. Here, the inner-loop reconfigurable control system identifies the changes in all system parameters affected by an elevon failure (there may be several). Now, the majority of RLV configurations employ body flaps for pitch trim control. In this case, a large portion of the pitch control function will typically be reverted to the body flaps. However, because body flaps are nominally used for to trim the vehicle, they are usually driven by much slower actuators than those associated with the elevons. Therefore, the resulting inner-closed-loop system will have a much lower bandwidth. At the guidance level, there is only *one* critical parameter associated with this example, namely the inner-loop bandwidth. Therefore, if the MSLS algorithm identifies a reduction in inner-loop bandwidth, the fundamental guidance reconfiguration strategy would be to reduce the feedback gains in the guidance loops to maintain stability margins. As another example, the net effect of headwinds, tailwinds, a stuck speedbrake, structural damage, etc. may all be lumped into one or two critical parameters, such as drag (or C_D) and/or lift-to-drag ratio.

Because of the project’s limited scope, the guidance loop reconfiguration and trajectory reshaping algorithms were developed in parallel, but not integrated here (they are shown integrated in Figure 1). Integration of these two elements is ongoing in current programs. For simplicity, the OPTG development can proceed without incorporating the adaptive guidance law. Once suitable OPTG performance is achieved, the adaptive guidance law can then be integrated to further improve the overall performance. Figure 2 presents this intermediate, modified OPTG architecture. It can be seen that the algorithm directly generates commands to the inner-closed-loop system. Although the guidance system runs open loop between inner-loop command updates, following these reconfigured commands implies that the vehicle will follow a reshaped trajectory.

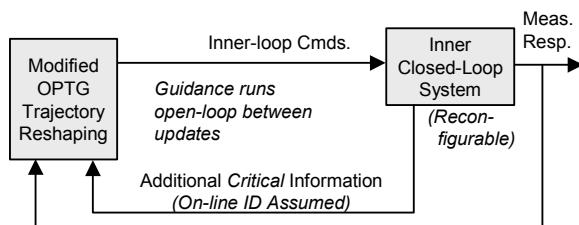


Figure 2. Intermediate OPTG Architecture.

Lastly, note that the OPTG development proceeded initially without on-line identification of the aforementioned *critical parameters*. In this case, the OPTG algorithm used the “true” values for these parameters. One would expect to sacrifice performance

when identified parameters are used, but to date, our experience has been that the OPTG algorithm is robust to reasonable parameter estimation errors. In related projects, studies of the MSLS identification algorithm’s performance for RLV applications have shown promising results (see, for example, [5]).

OPTG Trajectory Reshaping

Design Protocol - The Optimum-Path-To-Go (OPTG) guidance concept is based on the calculus of variations (COV), and has been successfully applied to numerous guidance and control applications, including "smart" munitions, unmanned air vehicles, high-agility military aircraft and most recently to related RLV work (e.g. [5], [9]). The OPTG methodology and the components that facilitate on-line trajectory reshaping are diagrammed in Figure 3 and described below.

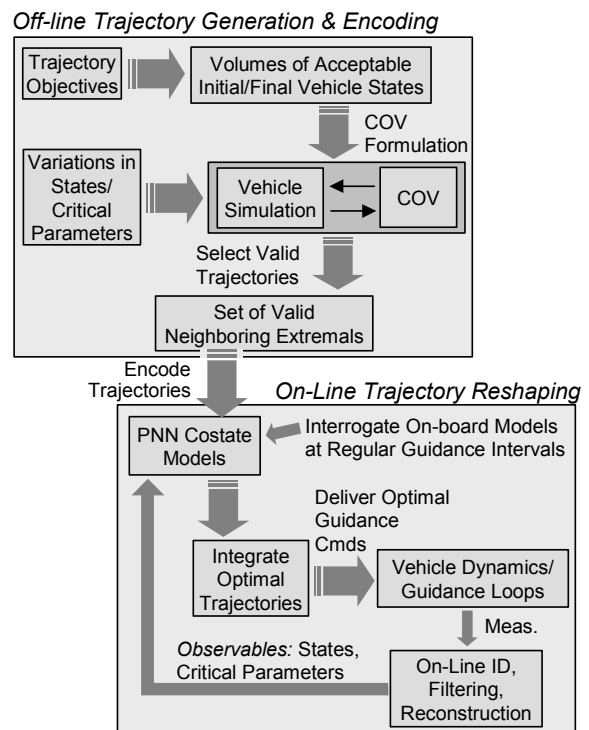


Figure 3. Off-Line and On-Line Components of the OPTG Approach.

Off-Line Trajectory Generation – The COV approach is used to formulate a two-point boundary value problem describing the trajectory optimization objectives. In the most general case of a nonlinear system, a numerical procedure is required to solve the two-point boundary value problem. This involves specifying all of the states and all of the costates (or Lagrange multipliers) at the same boundary so that numerical integration can be performed. We employed a common approach that involves an iterative numerical method or "shooting" technique to solve for the initial costate vector that

achieves the optimal trajectory. Once a single optimal trajectory is found, the steps taken to generate the OPTG database are as follows:

1. Perturb the specified initial states and repeat the iterative shooting procedure to create a library of neighboring optimal extremals. By taking small perturbations from the original optimal trajectory, subsequent solutions can be found in rapid fashion now that the shooting method starts from a good initial guess for the costate vector.
2. In-flight constraints are handled implicitly as part of the trajectory generation protocol. Control surface saturation, flight envelope boundaries, flight safety, structural integrity, and performance constraints, etc. are monitored during the course of the trajectory; integration is stopped and the trajectory is discarded if any of these conditions are encountered.
3. The trajectory generation protocol is then repeated for variations in identified *critical parameters* (such as lift or drag variations). From this, *libraries of libraries* of neighboring extremals are created. Such a 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 step provides a mechanism for manipulating the split boundary value problem into an initial value problem, and enables rapid on-line solutions of the OPTG problem.

From the libraries of optimal trajectories, one has state-costate pairs at all points along the numerous representative trajectories that are valid for the current mission. Using a nonlinear function-modeling tool, it is possible to find the mappings between the vehicle *observables* (states and critical parameters) and the costates at different points in the flight envelope. Polynomial neural networks (PNNs) are used for this task, which relate the observables to the costates for the libraries of trajectories. The result is efficient, compact, and, in general, nonlinear “table lookups” that parameterize the costates as functions of current vehicle observables.

On-Line Trajectory Integration – The final step in the OPTG approach is performed on-line. Trajectories are reshaped in flight to account for changes in the vehicle dynamics due to unmodeled atmosphere variations, aerodynamic uncertainties, or failures. During flight, the current vehicle states and critical parameters are obtained from measurements or reconstructed from the on-line identification algorithm; this information is then used to compute an appropriate set of costates from the on-board functional mappings (PNNs). With these states and co-states as starting values, the vehicle

dynamics and costate equations are integrated forward for a specified duration and used to compute the optimal commands. The loop is closed by re-initializing the costates at regular intervals – typically on the order of 1 Hz.

The OPTG methodology, as applied to the X-33 RLV case study will be discussed next.

COV Formulation - Again, the focus of the study was on the final approach-to-landing flight phase, which largely involves only vertical plane motion. Therefore, we limited the vehicle dynamics to a longitudinal model, and the states of the system were defined to be:

$$\bar{x} = [V, \gamma, X, H]^T \quad (1)$$

V = vehicle velocity ~ ft/sec
 γ = flight path angle ~ radians
 X = downrange position ~ feet
 H = altitude ~ feet

The OPTG algorithm was formulated to directly generate angle-of-attack commands to the inner-loop. Again, this was done to mimic the unmodeled guidance loops (see Figure 2). The complete formulation would involve integrating the OPTG algorithm with the adaptive guidance loops. There, the OPTG element would generate reference signals *into* the guidance loops (such as altitude, altitude rate and velocity commands – see Figure 1).

To build the pre-computed OPTG database, the calculus of variations (COV) [11,12] framework is used to generate optimal flight trajectories. The following optimization problem was proposed:

$$\text{Minimize: } J = \frac{1}{2} \int_{t_o}^{t_j} (\alpha - \alpha_{ref})^2 dt \quad (2)$$

Subject to:

$$\begin{aligned} \dot{V} &= \frac{-D}{m} - g \sin(\gamma) \\ \dot{\gamma} &= \frac{L}{mV} - \frac{g}{V} \cos(\gamma) \\ \dot{X} &= V \cos(\gamma) \\ \dot{H} &= V \sin(\gamma) \end{aligned} \quad (3)$$

(m = mass, g = grav. accel., D = drag, and L = lift)

$$\text{B.C.'s: } \begin{aligned} V(t_o) &= V_o & V(t_j) &= V_j \\ \gamma(t_o) &= \gamma_o & \gamma(t_j) &= \gamma_j \\ X(t_o) &= X_o & X(t_j) &= X_j \\ H(t_o) &= H_o & H(t_j) &= H_j \end{aligned} \quad (4)$$

$$C_3 X^3 + C_2 X^2 + C_1 X + C_0 - H = 0 \quad \text{for } X > X_j \quad (5)$$

Here, angle-of-attack, α , is defined as the commanded variable (generated by the OPTG algorithm, and delivered to the inner-closed-loop system – see Figure 2). The cost function defined above was chosen to minimize the difference between the commanded angle-of-attack and a reference angle-of-attack profile for a nominal X-33 approach trajectory. This cost function seemed to give the best results, given the problem formulation. By using a reference α profile, heritage design knowledge is implicitly folded into the cost function, and (at least at this design stage) eliminates the need to include more complex objectives. For a more detailed design, one may consider adding other terms in the cost function, such as a speed command penalty, as this vehicle does have some speed control capability.

The constraints of Eq. (3) are the governing equations of motion. The constraints in Eq. (4) with subscript o denote initial conditions, and with subscript j denote final conditions. The final conditions are not actually specified at the end of the mission (touchdown), but rather at an interior point defined by the downrange position X_j . Once at X_j , the vehicle is commanded to follow the equality constraint defined by Eq. (5) – discussed later.

The following simplified aerodynamic model was used in the COV formulation:

$$\begin{aligned} L &= \bar{q}SC_L, \quad C_L = 3.14\alpha + 0.0068 \\ D &= \bar{q}SC_D, \quad C_D = 1.40\alpha^2 + 0.018\alpha + 0.11 \quad (6) \\ \bar{q} &= \frac{1}{2}\rho(H)V^2 \end{aligned}$$

First, note that the Standard Atmosphere Model was used for obtaining air density values, $\rho(H)$. Second, C_L and C_D are the *trim* aerodynamic coefficients. The approximations given above were developed from a high fidelity X-33 aerodynamic model developed at the Air Force Research Laboratory (AFRL). This model provided the level of fidelity required for this problem – at this design stage. For one, scheduling the coefficients with Mach no. was not required at this low Mach flight phase. Furthermore, drag variations were studied here, and modeled by scaling C_D up or down in value. At more mature design stages, or addressing other scenarios involving particular control surface failures, additional fidelity may be required, such as scheduling the coefficients with certain parameters, or adding terms due to control surfaces deflections.

The unconstrained optimization problem is formulated by augmenting the cost function with the constraints of Eq. (3) using Lagrange multipliers. This ensures that the trajectories found by the calculus are feasible, (i.e., they obey the governing equations of motion).

Therefore, the following unconstrained optimization problem was proposed:

$$\text{Minimize:} \quad J = \int_{t_o}^{t_j} F(\bar{z}, \dot{\bar{z}}, t) dt \quad (7)$$

$$\text{Where:} \quad \bar{z} = [\bar{x}, \bar{\lambda}, \bar{u}]^T \quad (8)$$

$$\begin{aligned} F &= \frac{1}{2}(\alpha - \alpha_{ref})^2 + \lambda_1 \left[\dot{V} + \left(\frac{D}{m} + g \sin(\gamma) \right) \right] \\ &+ \lambda_2 \left[V\dot{\gamma} - \left(\frac{L}{m} - g \cos(\gamma) \right) \right] + \lambda_3 [\dot{X} - V \cos(\gamma)] \\ &+ \lambda_4 [\dot{H} - V \sin(\gamma)] \end{aligned} \quad (9)$$

The augmented vector, \bar{z} , consists of the state vector, define in Eq. (1), the costate vector (Lagrange multipliers), $\bar{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]^T$, and the control, $\bar{u} = \alpha$.

The general functions that optimize the augmented performance metric are determined by using the first variational necessary condition – commonly referred to as the *Euler equation* [11]:[†]

$$\text{for } t_o \leq t \leq t_j \quad \frac{d}{dt} \left(\frac{\partial F}{\partial \dot{\bar{z}}} \right) = \frac{\partial F}{\partial \bar{z}} \quad (10)$$

Applying Eq. (10) gives costate differential equations,

$$\begin{aligned} \dot{\lambda}_1 &= \lambda_1 \frac{D_V}{m} + \lambda_2 \left(\dot{\gamma} - \frac{L_V}{m} \right) - \lambda_3 \cos(\gamma) - \lambda_4 \sin(\gamma) \\ \dot{\lambda}_2 &= \lambda_1 \frac{g}{V} \cos(\gamma) - \lambda_2 \left(\frac{\dot{V} + g \sin(\gamma)}{V} \right) \\ &+ \lambda_3 \sin(\gamma) - \lambda_4 \cos(\gamma) \quad (11) \\ \dot{\lambda}_3 &= 0 \\ \dot{\lambda}_4 &= \lambda_1 \left(\frac{D_H}{m} \right) - \lambda_2 \left(\frac{L_H}{m} \right) \end{aligned}$$

$$\text{where } D_V = \frac{\partial D}{\partial V}, L_V = \frac{\partial L}{\partial V}, D_H = \frac{\partial D}{\partial H}, L_H = \frac{\partial L}{\partial H}.$$

Eq. (10) also gives the optimal command equation,

$$\alpha_{cmd} = \alpha_{ref} - \lambda_1 \left(\frac{D_\alpha}{m} \right) + \lambda_2 \left(\frac{L_\alpha}{m} \right) \quad (12)$$

$$\text{Likewise, } D_\alpha = \frac{\partial D}{\partial \alpha}, L_\alpha = \frac{\partial L}{\partial \alpha}.$$

[†] The Euler equation is equivalent to the *Euler-Lagrange equations* given in [12].

One challenge with the COV formulation is that significant drag reductions can result in solutions that exhibit large oscillatory motion at lower altitudes to manage the vehicle's energy. This can cause the vehicle to run into the ground prior to landing. *Inequality* constraints (e.g. $H > 0$ ft) are, in general, difficult to handle with the COV method. However, the altitude constraint can be met by specifying the interior point constraints given in Eq. (4) (at $t = t_j$), and the associated *equality* constraint given in Eq. (5). The interior point constraints force the vehicle to be at a specified state at a certain range from the runway, denoted X_j . For downrange positions beyond X_j , the equality constraint must be satisfied. By following this constraint, the altitude profile will follow a third order polynomial. The shape of this polynomial was chosen to bring the vehicle from the interior point to the desired touchdown point in a smooth and stable manner. This approach forces the energy management maneuvers (resulting from drag variations) to occur at less critical, higher altitudes – thus safeguarding the vehicle from running into the ground. For this case study, X_j was chosen to be $-3,280$ ft, which ensured that the oscillatory maneuvers occurred at sufficient altitudes. The coefficients of Eq. (5) are found by first recognizing that $\partial H/\partial X = \tan(\gamma)$. Then, evaluating both this relationship and Eq. (5) at the interior point and at touchdown gives:

$$\begin{bmatrix} H_j \\ H_f \\ \tan(\gamma_j) \\ \tan(\gamma_f) \end{bmatrix} = \begin{bmatrix} X_j^3 & X_j^2 & X_j & 1 \\ X_f^3 & X_f^2 & X_f & 1 \\ 3X_j^2 & 2X_j & 1 & 0 \\ 3X_f^2 & 2X_f & 1 & 0 \end{bmatrix} \begin{bmatrix} C_3 \\ C_2 \\ C_1 \\ C_0 \end{bmatrix} \quad (13)$$

Here, the subscript f refers to the targeted runway touchdown point. The flight path angle at touchdown was chosen to be -0.5 degrees. The flight path angle at the interior point and the altitude at the interior point and touchdown were all obtained from the nominal approach trajectory.

In summary, given a set of initializing costates, the equations of motion and costate equations are integrated forward to uniquely determine the optimal trajectory. At each control update, the optimal α -command is determined using Eq. (12). Integration is stopped at the interior point X_j . From this point to touchdown, the α -command is determined from the equality constraint given in Eq. (5). To solve for the α -command, the following second-order state variable equality constraint can be derived,

$$\begin{aligned} & [(3C_3X^2 + 2C_2X + C_1)\sin(\gamma) + \cos(\gamma)]G \\ & - (6C_3X + 2C_2)V \cos^2(\gamma) = 0 \end{aligned} \quad (14)$$

$$\text{where } G \equiv \left[\frac{L}{mV} - \left(\frac{g}{V} \right) \cos(\gamma) \right] \equiv \dot{\gamma} \quad (15)$$

This constraint is found by taking the first and second time derivatives of Eq. (5) and substituting in the expressions for \dot{X} , \dot{H} and $\dot{\gamma}$. This equation depends explicitly on α through lift, L (see Eq. (6)). From Eqs. (5) and (13), the altitude and flight path angle while on the equality constraint are given by,

$$\begin{aligned} H^* &= C_3X^3 + C_2X^2 + C_1X + C_0 \\ \gamma^* &= \tan^{-1}(3C_3X^2 + 2C_2X + C_1) \end{aligned} \quad (16)$$

Using these values in Eqs. (14) - (15) and solving for the α -command gives,

$$\alpha_{cmd} = \frac{mC_\gamma}{3.14\bar{q}S} ((6C_3X + 2C_2)V^2C_\gamma^2 + g) - 0.002 \quad (17)$$

where $C_\gamma = \cos(\gamma^*)$, the dynamic pressure, \bar{q} , is a function of H^* and S is the reference wing area. The numerical values in the above equation arise from the lift model given in Eq. (6).

It can be shown that continuity of the trajectory solution at the interior point X_j can be ensured if Eqs. (14) and (16) are also incorporated as boundary conditions (at $t = t_j$ - time at which the vehicle arrives at the interior point) in the COV formulation. Since this is a free final time problem, there are 5 independent variables - 4 costates and final time, t_j . Therefore, only 5 constraints can be satisfied - the 4 equations of motion and the equality constraint, Eq. (5). Because of this, the formulation is suboptimal because the transversality condition [11] at the final boundary has not been satisfied. Optimality could be obtained by assigning a 5th Lagrange multiplier associated with Eq. (5) and reformulating the problem. However, for this application the additional complexity associated with "optimality" was not required. Although suboptimal, this approach gives quite acceptable results.

For comparison, the optimal solution is found by satisfying the following transversality condition (rather than satisfying Eqs. (14) - (16)):

$$\text{for } t = t_j \left(\frac{\partial F}{\partial \bar{z}} \right)_{t_j}^T \delta \bar{z}_{t_j} + \left[F - \left(\frac{\partial F}{\partial \bar{z}} \right)_{t_j}^T \dot{\bar{z}} \right]_{t_j} \delta t_j \quad (18)$$

Here, $\delta \bar{z}$ denotes the first variation of \bar{z} and δt denotes the first variation of time, t . Again, for this free final time problem, δt_j is non-zero. Satisfying Eq. (18) leads

to the following solution for the optimal command at the interior point:

$$\alpha_{cmd} = \alpha_{ref} + \text{sqr}t\left(2(\lambda_1\dot{V} + \lambda_2\dot{\gamma} + \lambda_3\dot{X} + \lambda_4\dot{H})\Big|_{t_j}\right) \quad (19)$$

Figure 4 compares the optimal and suboptimal trajectories for a low drag case. The optimal trajectory satisfies the transversality condition, whereas the suboptimal trajectory satisfies the continuity condition at the interior point, X_j . For the optimal trajectory, the discontinuity in sink rate and angle-of-attack is evident. Therefore, as formulated, the optimal solution is unacceptable. On the other hand, it can be seen that there are no apparent severe penalties in using the suboptimal method, and the impact of ignoring the transversality condition seems to be minimal. Again, the benefit here is that the energy management maneuvers are done earlier in the flight, resulting in a more desirable altitude profile. Finally, note that because the equality constraint is followed at the end of the mission, the desired touchdown conditions are achieved for the suboptimal trajectory.

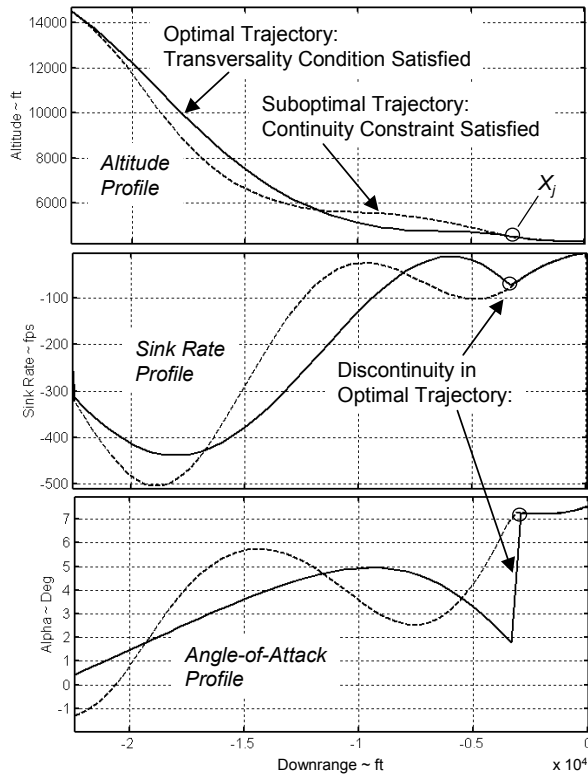


Figure 4. Comparison of Optimal and Suboptimal Trajectory Parameters.

Off-Line Trajectory Database Generation - Next, trajectory databases were generated by successively perturbing initial conditions and resolving the optimization problem. This method was used to

generate databases for variations in the states and in the drag coefficient, C_D , throughout the trajectory. Drag variations may be the consequence of mis-modeled aerodynamics, effector failures resulting in faulty speedbrake control, or large unexpected head/tail winds. Multiplicative errors were introduced in the value of C_D . It was found that the vehicle could recover from at least a 30% reduction to at least a 15% increase in C_D . Within this range, the vehicle was able to satisfy the nominal touchdown requirements, and the trajectories in each database included this full range of C_D . Each database was populated by first choosing a set of initial states at random from defined valid ranges. For each combination of initial states, a set of trajectories was generated for the entire range of C_D variations.

Recall that Polynomial Neural Networks (PNNs) are generated from these databases. The PNNs are interrogated at regular, discrete intervals on-line, corresponding to the starting points of the databases. In this work, the costate update interval was chosen to be = 1000 meters, or 3,280 ft. That is, PNNs are interrogated every 1000 meters downrange. A shorter distance between updates may be required for a more mature design, but this interval proved adequate for the proof-of-concept design presented here. The complete set of databases consists of approximately 6000 trajectories. Example parameters for the first two databases - trajectories initialized at 7000 m (22,960 ft) downrange and at 6000 m (19,680 ft) downrange - are shown in Figure 5.

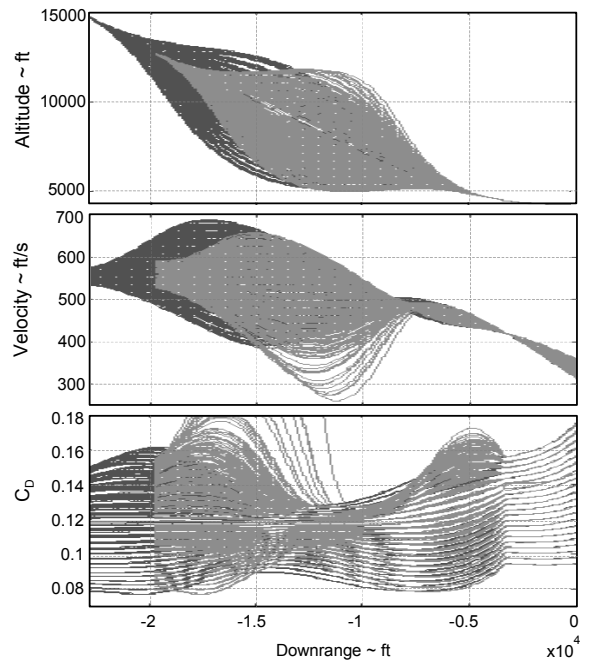


Figure 5. Databases for 23 K & 19K ft Downrange.

Off-Line Trajectory Encoding - Past experience in training PNNs to map states to costates revealed that model accuracy is typically increased as the input region is made smaller. For this reason, a separate PNN was trained for each update, rather than training one network for the entire mission segment. Therefore, the networks were trained using only trajectory data immediately surrounding each downrange update location.

It was found that the relationship between the states and costates for *increases* in drag was significantly different from the relationship for *decreases* in drag. This ill-conditioned behavior in the costates required each database to be further partitioned into low and high drag trajectories. A total of 14 databases were created for PNN synthesis (7 downrange locations/low and high drag variations). For each database, 4 PNNs were trained, one for each costate. All the PNNs generated for this experiment were found to be relatively simple networks and current processors should be able to easily handle their on-line computational burden.

On-Line Trajectory Reshaping - Attention is now turned to the on-line trajectory-reshaping element of the OPTG algorithm. As shown in Figure 3, estimates of the current states and critical parameters are used to interrogate the PNN models at regular guidance intervals. State estimates are obtained from filtered measurements and the critical parameters are obtained from the on-line identification algorithm, MSLS. Again, in this case study, the critical parameter was drag, or C_D . For this proof-of-concept development, the MSLS identification algorithm was mimicked by directly providing the OPTG algorithm with the drag variation information. The authors recognize that identification of drag can be a difficult task. However, some preliminary studies of this problem were performed in this program and showed promising results. We believe that accurate estimates of drag can be obtained from a standard set of vehicle state measurements.

With the costates obtained from the PNNs, the state and costate governing equations are then integrated forward to generate the angle-of-attack commands. This procedure is repeated every update interval. If changes in the drag coefficient are detected, this procedure will account for the change by reshaping the trajectory on-line to ensure that the desired landing conditions are achieved.

The PNNs were integrated into a C-code simulation model of the X-33 guidance system to provide costate updates during trajectory integration. Note that a simplified, low-order model was used to represent the "truth" model of the inner-closed-loop system. Development of a full 6-dof, high fidelity vehicle

model with an inner-loop reconfigurable control system was beyond the scope of the project. A Monte Carlo analysis was performed by introducing variations in vehicle drag. For a baseline comparison, these same variations were also introduced in a model of the system with only the production guidance law (i.e. no adaptation capabilities). A total of 1000 cases were defined for the Monte Carlo experiments. The variations are described as follows.

1. A random level of error in drag was introduced: The multiplicative error introduced to C_D was normally distributed with a mean of -7.5% and standard deviation of 6.0% . This kept the range of errors within 30% drag reductions to 15% drag increases (recall, it was determined that the vehicle was physically capable of achieving a safe landing if drag variations were within this range).
2. The drag error was introduced at a random downrange location in each trajectory. The downrange location where the change in drag occurred was normally distributed with a mean of 4000 m, and a standard deviation of 1,500 m.

Experimental Results

Baseline, Non-Adaptive Guidance Results - a histogram of the drag variations for the baseline case (production guidance system), is shown in Figure 6. This figure also indicates the number of trajectories that achieved successful landings and the number of unrecovered trajectories. A trajectory was considered to have achieved nominal touchdown conditions if the final values of the states were within a certain error from the nominal or desired final values. Touchdown values for velocity, flight path angle, downrange position, sink rate and angle-of-attack were all recorded and checked. It was found that the most critical parameter was the sink rate at touchdown. Missions that ended in touchdown sink rates less than 10 fps were considered acceptable. Note that the purpose of the baseline Monte Carlo studies was to show that the drag variations considered here were severe enough that the majority of missions would result in loss of the vehicle. To be conservative, we chose the value for an acceptable touchdown sink rate to be 10 fps. However, touching down at this sink rate may still cause severe structural damage to the vehicle. The baseline performance will degrade even further if tighter bounds on the touchdown conditions (e.g. sink rate no greater than, say, 7 fps) are defined.

The results in Figure 6 show that the production guidance system's performance is dismal at best. Nearly 71% of the missions failed to achieve acceptable touchdown conditions in the face of these sudden changes in drag. Drag reductions are seen to cause the

majority of the problems, as the vehicle is incapable of dissipating enough energy before the target touchdown point is reached. Figure 7 presents a histogram of the *downrange location* where the drag error was introduced. Here, un-recovered cases occur throughout the trajectory.

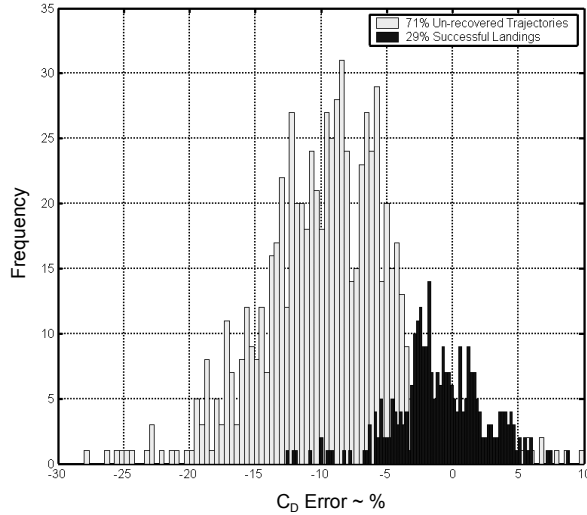


Figure 6. C_D Error Histogram (Non-Adaptive).

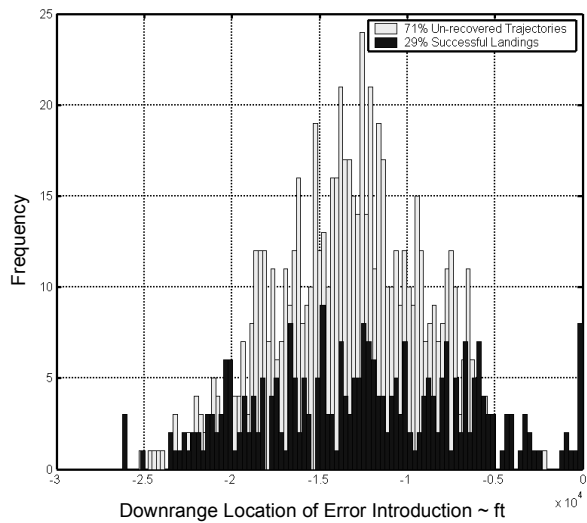


Figure 7. C_D Error Loc. Histogram (Non-Adaptive).

OPTG Trajectory Reshaping Results - A histogram of the drag variations is shown in Figure 8 for the system with the trajectory reshaping capabilities. It can be seen that the OPTG strategy was able to recover over 93% of the flights in the face of the sudden changes in drag. Likewise, a histogram of the downrange location where the drag error was introduced is shown in Figure 9. Here too, the excellent performance of the OPTG approach is indicated.

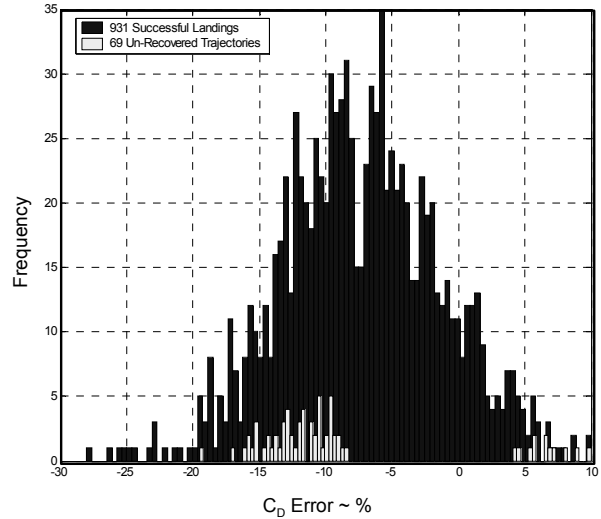


Figure 8. C_D Error Histogram (OPTG).

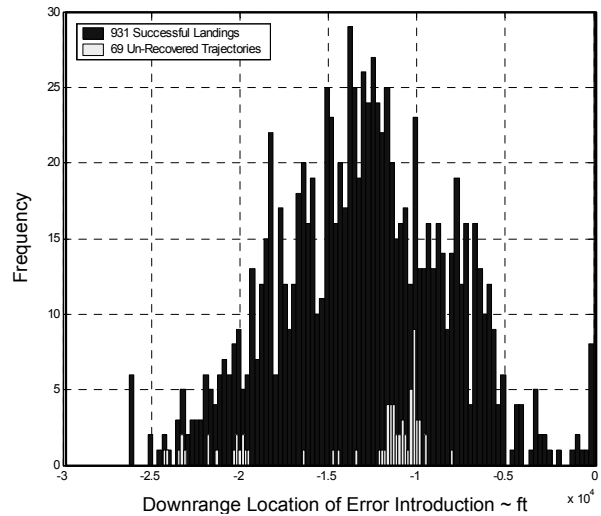


Figure 9. C_D Error Location Histogram (OPTG).

There may be several reasons for the missions that failed to achieve successful landings, including the following.

1. For the particular drag variations and “failure” locations, the PNNs were not trained on a sufficiently dense set of data to provide an accurate reshaped trajectory solution. This may be especially true for the distinct “cluster” of failures seen in Figure 8 (for the decreased drag cases). Increasing the PNN update rate may therefore eliminate many of these cases. This, of course, will require generation of a larger set of trajectory databases. Monte Carlo experiments such as this help to evolve the OPTG design paradigm. This result indicates that a tighter “grid” of PNNs may be required.

2. For the other “cluster” of mission failures, which occur at relatively high drag increases, the vehicle flew at angles-of-attack near or beyond the limits valid for the aerodynamic model ($-4 \text{ deg} < \alpha < 10 \text{ deg}$). This may have been a contributing factor to the unsuccessful missions.
3. The open-loop nature of this proof-of-concept approach does not take advantage of feedback in the guidance loops. Again, we mimicked a closed-loop guidance system by formulating the approach to generate angle-of-attack commands. Yet, the guidance system is actually flying open loop between each PNN update (a thousand meters!) The ultimate architecture of our proposed reconfigurable guidance system involves the integration of the OPTG trajectory reshaping with adaptive guidance feedback loops (see Figure 1). In that formulation, the OPTG algorithm would deliver reshaped altitude, altitude rate and velocity commands to the *closed-loop* adaptive guidance system. Experience has shown that this integrated architecture offers a significant further increase in robustness to variations in the identified critical parameters.

Nonetheless, the OPTG methodology was able to recover the vehicle and accomplish successful landings for an overwhelming number of the missions in the face of a wide range in drag variations and locations of the “failure.” For reductions in drag, the vehicle’s energy state was too high. The OPTG strategy was to perform Phugoid-like oscillations – trades in velocity and altitude – to manage the energy. For increases in drag, the vehicle’s energy state was too low. The OPTG strategy here was to turn up the flight path to gain needed downrange distance. This ensured that the vehicle had sufficient energy to arrive at the desired touchdown point. These characteristics are illustrated in Figure 10, which presents approach profiles for typical example low and high drag case studies.

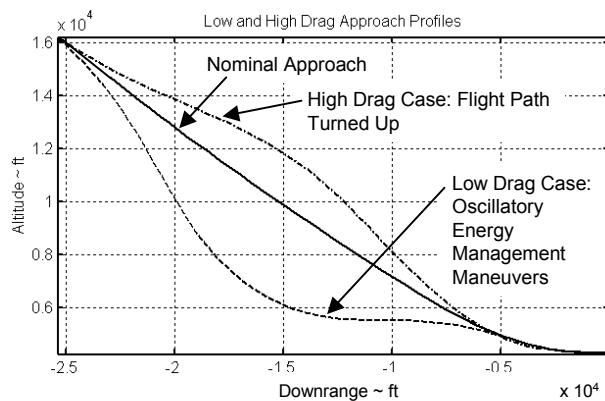


Figure 10. Comparison of Low & High Drag Approach Profiles.

Summary and Conclusions

NASA’s Space Launch Initiative (SLI) program has called for the development of on-line reconfiguration technologies to meet desired levels of reliability and safety. A reconfigurable control and guidance system for Reusable Launch Vehicles (RLVs) will consist of three main elements:

1. Control Reconfiguration: The inner-loop reconfigurable control system is tasked to maintain attitude stability and recover maneuvering performance to the extent possible in the face of rapid dynamic changes due to, for example, a catastrophic effector failure.
2. Guidance Reconfiguration: RLVs typically have a minimal set of control effectors, and full recovery to nominal maneuvering capabilities may be physically impossible for a large set of failure scenarios. The reconfigurable guidance system is tasked to follow reference trajectory commands and maintain flight stability in the face of degraded inner-closed-loop performance.
3. Trajectory Reshaping: Even though guidance law reconfiguration maintains flight stability, the final conditions of the mission segment may be unachievable if the trajectory commands driving the guidance loops are not reshaped on-line. For example, because of the vehicle’s degraded maneuvering capabilities, a reduction in commanded accelerations may be required, or energy management may call for a temporary change in the flight path to gain critical downrange distance.

Our approach calls for a system with all three of these elements. This paper concentrated on the trajectory-reshaping problem, and we have addressed this problem with the Optimum-Path-To-Go (OPTG) method. The focus of the studies was on the Approach/Landing flight phase, and Lockheed’s X-33 RLV served as the demonstration platform.

The fundamental idea behind the OPTG methodology is to pre-compute a database of neighboring extremals *off-line* that cover all variations under consideration. Then, using this data, polynomial-based networks are generated which map vehicle states and *critical parameters* (such as lift or drag) to optimal costates. The networks are then interrogated at regular intervals *on-line*, and the resulting costate information is used to integrate the solution forward and generate the optimal guidance commands. Therefore, based on the current states of the system and the current identified *critical parameters*, the algorithm will continually 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 due to variations in total vehicle drag. Such variations may come from mis-modeled aerodynamics, effector failures resulting in faulty speedbrake control, or large unexpected head or tail winds. The development addressed a *range* of both drag increases and decreases *anywhere* during the flight segment. In a Monte-Carlo analysis, one thousand trajectories were simulated, each with a random change in the drag at a random downrange location. The results of the study indicated that the OPTG method achieved excellent reconfiguration performance, with over 93% of the trajectories able to recover and safely land. Furthermore, note that for this proof-of-concept design, the guidance reconfiguration element was not included, and the system ran open loop between costate updates, which were every 1000 meters! The benefits of a closed-loop, adaptive guidance system should further improve the overall performance.

Acknowledgments

This work was funded by the Marshall Space Flight Center, Advanced Guidance & Control Program, Dr. John Hanson, Technical Representative. *Their support is gratefully appreciated.*

The authors would also like to acknowledge the help of Dr. David Doman of the Air Force Research Laboratory (AFRL), who supplied the aerodynamic model of the X-33 vehicle for this project.

References

1. 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.
2. Anon., "Introduction to NASA's Integrated Space Transportation Plan and Space Launch Initiative," NASA White Paper, May 17, 2001.
3. Lu, P., Shen, Z., Dukeman, G. A., Hanson, J. M., "Entry Guidance by Trajectory Regulation," AIAA-2000-3958, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August, 2000.
4. Johnson, E. N., Calise, A. J., Corban, J. E., "Reusable Launch Vehicle Guidance and Control Using Neural Networks," AIAA-2001-4381, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Montreal, August, 2001.
5. Schierman, J., Ward, D., Monaco, J., Hull, J., "A Reconfigurable Guidance Approach for Reusable Launch Vehicles," AIAA-2001-4429, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Montreal, August, 2001.
6. Pachter, M., P. Chandler, and M. Mears, "Reconfigurable tracking control with saturation," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, Sept. – Oct. 1995, pp. 1016 – 1022.
7. Rysdyk, R. and A. Calise, "Nonlinear adaptive flight control using neural networks," *IEEE Controls Systems Magazine*, Vol. 18, No. 6, Dec. 1998.
8. 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.
9. Ward, D., J. Monaco, and J. Schierman, "Reconfigurable control for VTOL UAV shipboard landing," AIAA Paper No. 99-4045, *Proc. AIAA Guidance, Navigation, and Control Conf.*, Portland, OR, Aug. 1999.
10. Hanson, J. M., "Advanced Guidance and Control Project for Reusable Launch Vehicles," AIAA-2000-3957, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August, 2000.
11. Kirk, D., *Optimal Control Theory, An Introduction*, Prentice-Hall Electrical Engineering Series, Englewood Cliffs, New Jersey, 1970.
12. Bryson, A., Ho, Y., *Applied Optimal Control*, Taylor & Francis, Bristol, PA, 1975.