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

New Nonlinear Control Technique for Ascent Phase of Reusable Launch Vehicles

David Drake,* Ming Xin,† and S. N. Balakrishnan‡
University of Missouri–Rolla, Rolla, Missouri 65401

Current flight control of reusable launch vehicles is based on table look-up gains for specific flight conditions. A new suboptimal nonlinear control technique for the complete ascent phase of reusable launch vehicles is presented. The new technique, called the *θ–D* method, is synthesized by adding perturbations to a typical optimal control formulation with a quadratic cost function. The controller expressions are obtained by getting an approximate closed-form solution to the Hamilton–Jacobi–Bellman equation. The *θ–D* method avoids iterative online solutions. A controller using this new method has been designed for the ascent phase of a reusable launch vehicle and implemented in a six-degrees-of-freedom high-fidelity simulator being run at the NASA Marshall Space Flight Center. Simulation results show that the *θ–D* controller achieves accurate tracking for the ascent phase of the reusable launch vehicle while being robust to external disturbances and plant uncertainties.

Nomenclature

$\boldsymbol{A}(x)$	=	state-dependent system matrix
$\boldsymbol{A}_0$	=	constant coefficient matrix used in θ–D method
$\boldsymbol{B}$	=	control matrix
$\boldsymbol{D}_i$	=	ith perturbation matrix in cost function of θ–D method
$\boldsymbol{g}(x)$	=	state-dependent control matrix
g_0	=	constant coefficient control matrix
$\boldsymbol{I}$	=	moment of inertia matrix
J	=	optimal control cost function
L, M, N	=	vehicle commanded torques (roll, pitch, yaw), ft · lb
$\boldsymbol{P}(x)$	=	feedback gain matrix
$\boldsymbol{Q}$	=	state weighting matrix
$\boldsymbol{R}$	=	control weighting matrix
$\boldsymbol{S}(\gamma)$	=	rotational matrix
$\boldsymbol{T}_i$	=	ith symmetric matrix in control solution of θ–D method
$\boldsymbol{u}$	=	control vector, L, M, and N
$\boldsymbol{x}$	=	vehicle state vector
γ	=	Euler angle vector, φ, θ, and ψ
θ	=	parameter used in the θ–D method
$\boldsymbol{\lambda}$	=	Lagrange's multiplier
τ	=	control torque commanded vector, L, M, and N
φ, θ, ψ	=	vehicle Euler angles, roll, pitch, and yaw, deg
$\boldsymbol{\omega}$	=	angular velocity vector, ω_x, ω_y, and ω_z
$\omega_x, \omega_y, \omega_z$	=	vehicle body rates, roll, pitch, and yaw, deg/s

Subscripts

s	=	sensed value
c	=	commanded value
x	=	component about the roll axis
y	=	component about the pitch axis
z	=	component about the yaw axis

Received 13 March 2004; revision received 6 July 2004; accepted for publication 7 July 2004.

*Graduate Student, Department of Mechanical and Aerospace Engineering, 1870 Miner Circle; dtdrake@umr.edu. Student Member AIAA.

†Postdoctoral Fellow, Department of Mechanical and Aerospace Engineering, 1870 Miner Circle; xin@umr.edu. Member AIAA.

‡Professor, Department of Mechanical and Aerospace Engineering, 1870 Miner Circle; bala@umr.edu. Associate Fellow AIAA.

I. Introduction

HANSON[1] details several past launch vehicle failure forms across the globe and the reasons for the test format of the NASA Marshall Spaceflight Center (MSFC) reusable launch vehicle (RLV) guidance and control development program. In another paper, Hanson discusses several methods that were being considered by NASA MSFC and their performance up to that time.[2] The flight control system for the X-33 RLV used a classical proportional–integral–derivative (PID) structure, where the controller parameters were obtained from table look-ups for specific flight conditions. The gains are scheduled over a set of operating points throughout the ascent flight profile. The breakpoints in the tables are chosen such that they encompass a range of Mach number, dynamic pressure, velocity, time, and any other important parameters. Gains are chosen at these select points and interpolated in the interval between the points. The baseline X-33 control structure consists of a PID controller using these scheduled gains. For the range of ascent cases, a total of 7800 values are needed in this PID control architecture.[3] A gain scheduling approach to control allocation is not very robust in the face of system changes or plant uncertainties. To handle changing flight conditions, disturbances, or vehicle uncertainties, separate off-line calculations of gain values need to be made and stored for look-up during the online run.

Because of the required design time of the gain-scheduled controller, new techniques have been investigated as alternatives to control the RLVs. An adaptive controller that uses dynamic inversion has been investigated by a group of researchers from Georgia Institute of Technology.[4] The method employed at Georgia Institute of Technology uses a neural network (NN) to compensate for the errors that occur from dynamic inversion. The adaptive controller using an NN has performed well throughout a range of flight conditions. Another approach being examined by Shtessel et al.[3,5,6] is control via a sliding mode controller. The sliding mode controller utilizes two loops that run at different timescales. Sliding mode is a good candidate for RLV ascent control because it is inherently robust. A drawback of the sliding mode approach is that a filter is required because of the highly oscillatory nature of the results it generates. Trajectory linearization[7] has also been examined for its merits as a flight controller on RLVs. Doman and Ngo[8] have investigated the use of direct and indirect adaptive control system for an X-33 vehicle. Hodel and Callahan[9] have carried out some important studies on the reassignment of controller efforts for RLVs.

An optimal or suboptimal control formulation reduces the need for multiple off-line runs and gives the added advantage of

robustness of the controller in the face of external disturbances and plant uncertainties. It has been shown theoretically for linear systems that under some restrictions an optimal controller can tolerate 6-dB gain reductions and 60-deg phase margin[10] when using the linear-quadratic-regulator problem formulation. Although it is difficult to extend these stability margin concepts rigorously to nonlinear systems, it has been observed that optimality-based controllers are robust to a high level of model and parametric uncertainties.[11] One of the major difficulties of an optimal control formulation in nonlinear problems is that no analytic solution exists in general. Some approximate or suboptimal control techniques have been proposed in the literature. One approach in particular is the state-dependent Riccati equation (SDRE) method (see Ref. 12). The SDRE technique is able to solve the controller design for nonlinear systems by solving the algebraic Riccati equation at every time step. However, this poses an implementation problem because solving Riccati equations online requires a large amount of online computational power. In the current study, which has close ties to the SDRE technique, there is no need for such iterative computations.

In this paper, the θ–D control design[13] is derived by making modifications to a quadratic performance-index-based cost function to be minimized. These modifications enable one to obtain a closed-form solution for the θ–D control law that is easy to implement in any onboard computer and obviates the need for a large amount of online computations. The θ–D method has been successfully applied to a nonlinear missile autopilot design.[14,15]

The rest of the paper is organized as follows. The θ–D control technique is developed in Sec. II. Vehicle dynamics used in the controller development are described in Sec. III. Controller design is discussed in Sec. IV. Simulation results are analyzed in Sec. V. Conclusions are presented in Sec. VI.

II. Summary of θ–D Suboptimal Control Method

The θ–D control technique produces a nonlinear feedback control law that is able to control accurately and robustly highly nonlinear systems. The θ–D synthesis[13] was formulated for state feedback control of a class of nonlinear regulator problems. The time-invariant control-affine system for which this technique is valid is described by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \tag{1}$$

A stabilizing controller for the system in Eq. (1) can be synthesized by minimizing the quadratic cost function,

$$J = \frac{1}{2}\int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\,\mathrm{d}t \tag{2}$$

where $\boldsymbol{x} \in R^n$, $\boldsymbol{f} \in R^n$, $\boldsymbol{B} \in R^{n\times m}$, $\boldsymbol{u} \in R^m$, $\boldsymbol{Q} \in R^{n\times n}$, $\boldsymbol{R} \in R^{m\times m}$, and $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$. $\boldsymbol{Q}$ is semipositive definite, and $\boldsymbol{R}$ is positive definite. An assumption that $\boldsymbol{x} \in \Omega_x$ and Ω_x is a compact set in R^n is required for this method. The solution to the optimal control problem in Eqs. (1) and (2) is assumed to exist and to be unique.

The optimal solution of the infinite-horizon nonlinear regulator problem can be obtained by solving the Hamilton–Jacobi–Bellman (HJB) partial differential equation (see Ref. 16),

$$\frac{\partial V^T}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) - \frac{1}{2}\frac{\partial V^T}{\partial \boldsymbol{x}}\boldsymbol{B}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{x})\frac{\partial V}{\partial \boldsymbol{x}} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} = 0 \tag{3}$$

where $V(\boldsymbol{x})$ is the optimal cost, that is,

$$V(\boldsymbol{x}) = \min_u \frac{1}{2}\int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\,\mathrm{d}t \tag{4}$$

where $V(\boldsymbol{x}) > 0$, $V(\boldsymbol{0}) = 0$, and $V(\boldsymbol{x})$ is assumed to be continuously differentiable.

The expression for the optimal control is given by

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{x})\frac{\partial V}{\partial \boldsymbol{x}} \tag{5}$$

The HJB equation is extremely difficult to solve in general, which renders optimal control techniques of limited use for nonlinear systems. To develop an approximate series solution to this problem, perturbations are added to the cost function, which transforms Eq. (2) into

$$J = \frac{1}{2}\int_0^\infty \left[\boldsymbol{x}^T\left(\boldsymbol{Q} + \sum_{i=1}^{\infty}\boldsymbol{D}_i\theta^i\right)\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}\right]\mathrm{d}t \tag{6}$$

where θ and $\boldsymbol{D}_i$ are chosen such that

$$\boldsymbol{Q} + \sum_{i=1}^{\infty}\boldsymbol{D}_i\theta^i$$

is semi-positive definite.

Rewriting the original state equation in a linearlike form gives

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} = \{\boldsymbol{A}_0 + \theta[\boldsymbol{A}(\boldsymbol{x})/\theta]\}\boldsymbol{x} + \{\boldsymbol{g}_0 + \theta[\boldsymbol{g}(\boldsymbol{x})/\theta]\}\boldsymbol{u} \tag{7}$$

where $\boldsymbol{A}_0$ is a constant matrix such that $(\boldsymbol{A}_0, \boldsymbol{g}_0)$ is a stabilizable pair and $\{[\boldsymbol{A}_0 + \boldsymbol{A}(\boldsymbol{x})], [\boldsymbol{g}_0 + \boldsymbol{g}(\boldsymbol{x})]\}$ is pointwise controllable. Here θ is an intermediate variable for the purpose of power series expansion. When

$$\boldsymbol{\lambda} = \frac{\partial V}{\partial \boldsymbol{x}} \tag{8}$$

is defined and Eq. (8) and Eq. (7) are substituted into Eq. (3), the perturbed HJB equation is

$$\lambda^T\boldsymbol{f}(\boldsymbol{x}) - \frac{1}{2}\lambda^T\boldsymbol{B}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{x})\lambda + \frac{1}{2}\boldsymbol{x}^T\left[\boldsymbol{Q} + \sum_{i=1}^{\infty}\boldsymbol{D}_i\theta^i\right]\boldsymbol{x} = 0 \tag{9}$$

A power series expansion of $\boldsymbol{\lambda}$ in terms of θ is assumed,

$$\boldsymbol{\lambda} = \frac{\partial V}{\partial \boldsymbol{x}} = \sum_{i=0}^{\infty}\boldsymbol{T}_i\theta^i\boldsymbol{x} \tag{10}$$

with the $\boldsymbol{T}_i$ being symmetric matrices. Substituting Eq. (10) into Eq. (9) and equating the coefficients of powers of θ to zero yields

$$\boldsymbol{T}_0\boldsymbol{A}_0 + \boldsymbol{A}_0^T\boldsymbol{T}_0 - \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0 + \boldsymbol{Q} = 0 \tag{11}$$

$$\boldsymbol{T}_1\left(\boldsymbol{A}_0 - \boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0\right) + \left(\boldsymbol{A}_0^T - \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\right)\boldsymbol{T}_1 = -\frac{\boldsymbol{T}_0\boldsymbol{A}(\boldsymbol{x})}{\theta} - \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_0}{\theta} + \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_0 + \boldsymbol{T}_0\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0 - \boldsymbol{D}_1 \tag{12}$$

$$\boldsymbol{T}_2\left(\boldsymbol{A}_0 - \boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0\right) + \left(\boldsymbol{A}_0^T - \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\right)\boldsymbol{T}_2 = -\frac{\boldsymbol{T}_1\boldsymbol{A}(\boldsymbol{x})}{\theta} - \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_1}{\theta} + \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_1 + \boldsymbol{T}_0\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_1 + \boldsymbol{T}_0\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_0 + \boldsymbol{T}_1\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_1 + \boldsymbol{T}_1\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_0 + \boldsymbol{T}_1\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0 - \boldsymbol{D}_2 \tag{13}$$

$$\boldsymbol{T}_n\left(\boldsymbol{A}_0 - \boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0\right) + \left(\boldsymbol{A}_0^T - \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\right)\boldsymbol{T}_n = -\frac{\boldsymbol{T}_{n-1}\boldsymbol{A}(\boldsymbol{x})}{\theta} - \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_{n-1}}{\theta} + \sum_{j=0}^{n-2}\boldsymbol{T}_j\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_{n-2-j} + \sum_{j=0}^{n-1}\boldsymbol{T}_j\left[\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta} + \frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\right]\boldsymbol{T}_{n-1-j} + \sum_{j=1}^{n-1}\boldsymbol{T}_j\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_{n-j} - \boldsymbol{D}_n \tag{14}$$

Note that Eq. (11) consists of constant coefficient matrices and is an algebraic Riccati equation. Equations (12–14) are Lyapunov equations that are linear in terms of $\boldsymbol{T}_i$, $i = 1, \ldots, n$. Because the right-hand side of Eqs. (12–14) involves $\boldsymbol{x}$ and θ, $\boldsymbol{T}_i$ is a function of $\boldsymbol{x}$ and θ. Thus, we denote it as $\boldsymbol{T}_i(\boldsymbol{x}, \theta)$. The expression for control can be obtained in terms of a power series as

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{x})\frac{\partial V}{\partial \boldsymbol{x}} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{x})\sum_{i=0}^{\infty}\boldsymbol{T}_i(\boldsymbol{x},\theta)\theta^i\boldsymbol{x} \tag{15}$$

Expressions for the perturbation matrices are constructed using the following pattern:

$$\boldsymbol{D}_1 = k_1 e^{-a_1 t}\left[-\frac{\boldsymbol{T}_0\boldsymbol{A}(\boldsymbol{x})}{\theta} - \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_0}{\theta} + \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_0 + \boldsymbol{T}_0\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0\right] \tag{16}$$

$$\begin{aligned}\boldsymbol{D}_2 = k_2 e^{-a_2 t}\Bigg[&-\frac{\boldsymbol{T}_1\boldsymbol{A}(\boldsymbol{x})}{\theta} - \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_1}{\theta} + \boldsymbol{T}_0\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_1 \\ &+ \boldsymbol{T}_0\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_1 + \boldsymbol{T}_0\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_0 + \boldsymbol{T}_1\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_0 \\ &+ \boldsymbol{T}_1\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_0 + \boldsymbol{T}_1\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_1\Bigg]\end{aligned} \tag{17}$$

$$\vdots$$

$$\begin{aligned}\boldsymbol{D}_n = k_n e^{-a_n t}\Bigg\{&-\frac{\boldsymbol{T}_{n-1}\boldsymbol{A}(\boldsymbol{x})}{\theta} - \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_{n-1}}{\theta} \\ &+ \sum_{j=0}^{n-1}\boldsymbol{T}_j\left[\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta} + \frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\right]\boldsymbol{T}_{n-1-j} \\ &+ \sum_{j=0}^{n-2}\boldsymbol{T}_j\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_{n-2-j} + \sum_{j=1}^{n-1}\boldsymbol{T}_j\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_{n-j}\Bigg\}\end{aligned} \tag{18}$$

where k_i and $a_i > 0$, $i = 1, \ldots, n$ are adjustable design parameters.

The idea in constructing $\boldsymbol{D}_i$ in this manner is based on the observation that large initial states may give rise to large initial control due to the state-dependent term $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ on the right-hand side of Eqs. (12–14). A large initial control happens when there are some terms in $\boldsymbol{A}(\boldsymbol{x})$ or $\boldsymbol{g}(\boldsymbol{x})$ that could grow to a large magnitude if the initial $\boldsymbol{x}$ is large. For example, when $\boldsymbol{A}(\boldsymbol{x})$ includes a cubic term, its magnitude could be large if $\boldsymbol{x}$ is large. A large value in $\boldsymbol{A}(\boldsymbol{x})$ will be reflected into the solution for $\boldsymbol{T}_i$, that is, the left-hand side of Eqs. (12–14). Because $\boldsymbol{T}_i$ will be used in the next equation to solve for $\boldsymbol{T}_{i+1}$, this large value will be propagated and amplified and, consequently, command larger control inputs or even instability. Thus, if $\boldsymbol{D}_i$ is chosen such that

$$\begin{aligned}&-\frac{\boldsymbol{T}_{i-1}\boldsymbol{A}(\boldsymbol{x})}{\theta} - \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_{i-1}}{\theta} + \sum_{j=0}^{i-2}\boldsymbol{T}_j\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_{i-2-j} \\ &+ \sum_{j=0}^{i-1}\boldsymbol{T}_j\left(\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta} + \frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\right)\boldsymbol{T}_{i-1-j} \\ &+ \sum_{j=1}^{i-1}\boldsymbol{T}_j\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_{i-j} - \boldsymbol{D}_i = \varepsilon_i(t)\Bigg\{-\frac{\boldsymbol{T}_{i-1}\boldsymbol{A}(\boldsymbol{x})}{\theta} \\ &- \frac{\boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_{i-1}}{\theta} + \sum_{j=0}^{i-1}\boldsymbol{T}_j\left[\boldsymbol{g}_0\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta} + \frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\right]\boldsymbol{T}_{i-1-j} \\ &+ \sum_{j=0}^{i-2}\boldsymbol{T}_j\frac{\boldsymbol{g}(\boldsymbol{x})}{\theta}\boldsymbol{R}^{-1}\frac{\boldsymbol{g}^T(\boldsymbol{x})}{\theta}\boldsymbol{T}_{i-2-j} + \sum_{j=1}^{i-1}\boldsymbol{T}_j\boldsymbol{g}_0\boldsymbol{R}^{-1}\boldsymbol{g}_0^T\boldsymbol{T}_{i-j}\Bigg\}\end{aligned} \tag{19}$$

where

$$\varepsilon_i(t) = 1 - k_i e^{-a_i t} \tag{20}$$

is a small number, ε_i can be used to suppress this large value from propagating in Eqs. (12–14). In summary, there are three functions for ε_i. The first usage is to prevent the occurrence of large values of control. The second function is to satisfy some conditions required in the proof of convergence and stability of the θ–D algorithm.[13] The third usage is to modulate the system transient performance by adjusting the parameters of k_i and a_i. On the other hand, the exponential term $e^{-a_i t}$ with $a_i > 0$ is also used to let the perturbation terms in the cost function and HJB equation diminish as time evolves. A systematic method has been developed to determine the $\boldsymbol{D}_i$ terms. This systematic approach is included in the Appendix.

Remark 1: The steps of applying the θ–D method are summarized as follows.

1) Solve the algebraic Riccati equation (11) to get $\boldsymbol{T}_0$ once $\boldsymbol{A}_0, \boldsymbol{g}_0, \boldsymbol{Q}$, and $\boldsymbol{R}$ are determined. Note that the resulting $\boldsymbol{T}_0$ is a positive-definite constant matrix.

2) Solve the linear Lyapunov equation (12), to obtain $\boldsymbol{T}_1(x, \theta)$ as a closed-form solution.

3) Solve Eqs. (13) and (14) by following the same procedure in step 2. Note that the number of $\boldsymbol{T}_i$ needed depends on the problem. The simulation results show that $\boldsymbol{T}_0$, $\boldsymbol{T}_1$, and $\boldsymbol{T}_2$ are sufficient to achieve satisfactory performance for this problem.

Closed-form solutions for $\boldsymbol{T}_1, \ldots, \boldsymbol{T}_n$ can be obtained with just one matrix inverse operation. Because $\boldsymbol{T}_1, \ldots, \boldsymbol{T}_n$ are obtained in this manner, no iterative online calculations are necessary. The introduction of θ is for the convenience of power series expansion, and θ gets canceled when $\boldsymbol{T}_i(\boldsymbol{x}, \theta)$ multiplies θ^i in the final control calculations, that is, Eq. (15). Note that in every $\boldsymbol{T}_i$ equation there is a $1/\theta^i$ factor on the right-hand side of the equation. Because of linearity of the $\boldsymbol{T}_i$ equation, $1/\theta^i$ will appear in the solution of $\boldsymbol{T}_i$, that is, $\boldsymbol{T}_i = (1/\theta^i)\hat{\boldsymbol{T}}_i$ where $\hat{\boldsymbol{T}}_i$ is the solution without $1/\theta^i$. When $\boldsymbol{T}_i$ is multiplied by θ^i in the controller Eq. (15), θ^i is canceled.

Remark 2: Because the θ–D method stems from a quadratic cost function, a comparison with the linear-quadratic-regulator (LQR) is in order. The major advantage of LQR is that the controller design is straightforward and easy to implement. However, it is not likely that a single LQR controller would be able to handle the entire range of ascent for the RLV due to the nonlinearities associated with the system dynamics. However, a series of LQR controllers over discrete intervals could provide good results with linearized dynamics. Two problems will arise when implementing the LQR solution to the ascent problem. The first is to determine a set of acceptable operating points about which linearization is performed without resulting in a model that is not true to the physics of the system. Second, although the LQR has constant gains and, therefore, is computationally less complex, gain scheduling the LQR for different intervals and the design of those intervals may make the design process as laborious as the conventional PID controller design. In contrast, the θ–D method deals with nonlinear dynamic model directly. The major advantage of the closed-form feedback controller provided by this method will make this nonlinear controller computationally easy to implement. Additionally simulation will show that one set of well-tuned parameters for the θ–D controller works out over the whole ascent phase.

III. Equations of Motion for the RLV

Modeling the RLV equations of motion was desired to be kept simple while capturing the physics of the problem to ensure good results. The dynamics equation[5] that governs the rotational motion of the rigid body RLV is given by

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\Omega}\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{\tau} \tag{21}$$

where $\boldsymbol{I} \in R^{3\times 3}$, $\boldsymbol{\omega} = [\omega_x \;\; \omega_y \;\; \omega_z]^T$, and $\boldsymbol{\tau} = [L \;\; M \;\; N]^T$. $\boldsymbol{\Omega}$ is the matrix given by

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{22}$$

The $\boldsymbol{\tau}$ in Eq. (21) includes contributions from the control effectors only. Wing body aerodynamic contributions to vehicle moments are not included in the dynamics of this study because these effects are proportional to the body-axis rates, which are small during ascent, (shown subsequently). Also of note in Eq. (21) is the assumption that $\dot{\boldsymbol{I}} = 0$. This simplification is commonly used in RLV applications.[5,8] The simulation results in Sec. V show that these unmodeled dynamics do not cause problems with the Euler angle tracking.

The relationship between the body rates and Euler angles are described by

$$\dot{\gamma} = \boldsymbol{S}(\gamma)\boldsymbol{\omega} \tag{23}$$

where $\gamma = [\varphi \;\; \theta \;\; \psi]^T$ and $\boldsymbol{S}(\gamma)$ is denoted as

$$\boldsymbol{S}(\gamma) = \begin{bmatrix} 1 & \tan\theta\sin\varphi & \tan\theta\cos\varphi \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \dfrac{\sin\varphi}{\cos\theta} & \dfrac{\cos\varphi}{\cos\theta} \end{bmatrix} \tag{24}$$

The rotation sequence in Eq. (24) that is used in this work was taken from the work in Ref. 5. When Eq. (24) is examined, it is seen that $\tan(\theta)$ appears. This can cause a singularity to occur at the beginning of the ascent phase when θ is 90 deg. A reformulation of the system to the plumbline axis system or quaternions to eliminate this singularity was not done because handling the physics using Euler angles is more natural than using quaternions. Whereas other control methods would require this reformulation, the θ–D control method is able to handle the singularity with the $\boldsymbol{D}_i$ terms. The $\boldsymbol{D}_i$ terms are able to overcome the large initial controls that occur when $\tan(\theta)$ is large, as is indicated in Sec. II. For these reasons, it was decided to allow the controller to handle the singularity.

Again, note that only the rotational motion is considered in the controller design. The desired control-induced moments are passed on to the control allocation scheme in the six-degrees-of freedom (6-DOF) high-fidelity X33 simulator where the complete rotational as well as the translational motions are considered. Details of the simulation are presented in Sec. V.

IV. θ–D Controller Design

The dynamics in Eqs. (21) and (23) were used in the θ–D approximation method to develop a control law. The objective of the controller design is to track the desired trajectories of the Euler angles while minimizing the control energy. To design the controller to perform command following, the θ–D controller is implemented as an integral servomechanism[15] to improve the tracking accuracy and robustness to disturbances and uncertainties. Therefore, integral states are incorporated in the state space. A set of nine states, Euler angles, angular rates, and the integral of Euler angles, was used to form the elements of the state space, which is

$$\boldsymbol{x} = \left[\int \varphi_s \quad \int \theta_s \quad \int \psi_s \quad \omega_x \quad \omega_y \quad \omega_z \quad \varphi_s \quad \theta_s \quad \psi_s \right]^T \tag{25}$$

The θ–D integral controller is given by

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}(\boldsymbol{x})^T \sum_{n=0}^{\infty} \boldsymbol{T}_n(\boldsymbol{x},\theta)\theta^n \left[\int \varphi_s - \int \varphi_c \quad \int \theta_s - \int \theta_c \right.$$
$$\left. \int \psi_s - \int \psi_c \quad \omega_x \quad \omega_y \quad \omega_z \quad \varphi - \varphi_c \quad \theta_s - \theta_c \quad \psi_s - \psi_c \right]^T \tag{26}$$

The subscript c denotes a commanded value, and the subscript s denotes a sensed or actual value. Previous applications of the θ–D technique[14,15] have shown that using the first two perturbation terms in Eq. (15) is sufficient for attaining good results. For this reason, two perturbation terms were used for the given RLV problem. To apply the θ–D technique, a problem of the form of Eq. (1) is required. Upon examination of Eq. (21), the given dynamics are properly suited for the θ–D method.

For use in the θ–D technique, the nonlinear Eq. (7) was partitioned in the following way:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} = \{\boldsymbol{A}_0 + \theta[\boldsymbol{A}(\boldsymbol{x})/\theta]\}\boldsymbol{x}$$
$$+ \{\boldsymbol{g}_0 + \theta[\boldsymbol{g}(\boldsymbol{x})/\theta]\}\boldsymbol{u} \tag{27}$$

where $\boldsymbol{B}(\boldsymbol{x}) \in R^{9x3}$ is a state-dependent matrix, $\boldsymbol{F}(\boldsymbol{x}) \in R^{9x9}$ is a state-dependent matrix, and $\boldsymbol{u} \in R^{3x1}$ is the control vector of torques. $\boldsymbol{F}(\boldsymbol{x})$, $\boldsymbol{B}(\boldsymbol{x})$, and $\boldsymbol{u}$ are

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \mathrm{diag}(1,\,1,\,1) \\ \boldsymbol{0}_{3\times 3} & -\boldsymbol{I}(\boldsymbol{t})^{-1}\boldsymbol{\Omega}\boldsymbol{I}(\boldsymbol{t}) & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{S}(\gamma) & \boldsymbol{0}_{3\times 3} \end{bmatrix} \tag{28}$$

$$\boldsymbol{B} = [\boldsymbol{0}_{3\times 3} \quad \boldsymbol{I}(\boldsymbol{t})^{-1} \quad \boldsymbol{0}_{3\times 3}]^T \tag{29}$$

$$\boldsymbol{u} = [L \quad M \quad N]^T \tag{30}$$

In this problem, we choose the factorization of nonlinear Eq. (27) as

$$\dot{\boldsymbol{x}} = (\boldsymbol{F}(\boldsymbol{x}_0) + \theta\{[\boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{F}(\boldsymbol{x}_0)]/\theta\})\boldsymbol{x} + (\boldsymbol{B}(\boldsymbol{x}_0)$$
$$+ \theta\{[\boldsymbol{B}(\boldsymbol{x}) - \boldsymbol{B}(\boldsymbol{x}_0)]/\theta\})\boldsymbol{u} \tag{31}$$

The advantage of choosing this factorization is that in the θ–D formulation $\boldsymbol{T}_0$ is solved from $\boldsymbol{A}_0$ and $\boldsymbol{g}_0$ in Eq. (11). By the selection of $\boldsymbol{A}_0 = \boldsymbol{F}(\boldsymbol{x}_0)$ and $\boldsymbol{g}_0 = \boldsymbol{B}(\boldsymbol{x}_0)$, a good starting point for $\boldsymbol{T}_0$ is obtained because $\boldsymbol{F}(\boldsymbol{x}_0)$ and $\boldsymbol{B}(\boldsymbol{x}_0)$ retain much more system information than an arbitrary choice of $\boldsymbol{A}_0$ and $\boldsymbol{g}_0$ would.

An optimal control cost function of

$$J = \frac{1}{2}\int_0^{\infty} \left\{ \boldsymbol{x}^T \left[\boldsymbol{Q} + \boldsymbol{D}_1\theta + \boldsymbol{D}_2\theta^2 \right]\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} \right\} \mathrm{d}t \tag{32}$$

was used where $\boldsymbol{Q} \in R^{9\times 9}$ is diagonal matrix, $\boldsymbol{R} \in R^{3\times 3}$, $\boldsymbol{D}_1 \in R^{9\times 9}$, $\boldsymbol{D}_2 \in R^{9\times 9}$, and θ is a scalar value. As mentioned in Sec. II, the control expression consisted of the first three terms of the series as

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}(\boldsymbol{x})\boldsymbol{x} \tag{33}$$

where $\boldsymbol{P}(\boldsymbol{x})$ is

$$\boldsymbol{P}(\boldsymbol{x}) = \boldsymbol{T}_0 + \boldsymbol{T}_1(\boldsymbol{x},\theta)\theta + \boldsymbol{T}_2(\boldsymbol{x},\theta)\theta^2 \tag{34}$$

The controller weighting matrix $\boldsymbol{R}$ is of particular interest in the controller design. A series of static values for the $\boldsymbol{R}$ matrix were tried, with $\boldsymbol{R}$ ranging from 10^{-13} up through 10^{13}. Values of 10^{-12} were determined to produce the best results from the controller. The optimal control formulation was examined to see if some form of dependence could be found to rationalize this low value. In looking at the closed-loop state equation

$$\dot{\boldsymbol{x}} = [\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}(\boldsymbol{x})]\boldsymbol{x} \tag{35}$$

and examining Eq. (33), we would like to offset the effect of the $\boldsymbol{B}$ and $\boldsymbol{B}^T$ so that $\boldsymbol{P}(\boldsymbol{x})$ is not multiplied by an extremely large or extremely small number. The $\boldsymbol{B}$ matrix is a function of the inverse of the moments of inertia, which are on the order of 10^{-6}. Thus, to offset the $\boldsymbol{B}$ multiplying $\boldsymbol{B}^T$ that is on the order of 10^{-12}, the $\boldsymbol{R}^{-1}$ needs to be on the order of 10^{12}. Also, to better track the physics of the launch vehicle, the $\boldsymbol{R}$ matrix was scheduled as a function of $\boldsymbol{I}$ and used to set $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T$ to unity in all elements. Scheduling the $\boldsymbol{R}$ matrix in this manner was very helpful because the mass properties

of a launch vehicle vary greatly from launch to orbital insertion due to fuel consumption. If the $\boldsymbol{R}$ matrix is set to a constant value based on an inertia matrix value from a certain time, oscillations would occur in the Euler angles. By setting $\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{T}$ to unity and making $\boldsymbol{R}$ a function of $\boldsymbol{I}$, we overcame the oscillation problem, and the numerical results showed our selection to be effective.

The $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ terms were also examined over a range of possible values. Values for the k_1 and k_2 terms were chosen to be 1, which helps to avoid the generation of large initial control commands when some initial states are large.[13] The exponent values, a_1 and a_2, determine how quickly the $\boldsymbol{D}_i$ term dies away, with a large value reducing the $\boldsymbol{D}_i$ term to zero quickly and a small value reducing the value slowly. For example, an a_1 of 10 reduces $k_1 e^{-a_1 t}$ to zero in 0.4 s, whereas an a_1 of 0.0001 takes 40,000 s to reduce $k_1 e^{-a_1 t}$ to zero. Again, the physics of the problem were used to determine values for a_1 and a_2 in the $\boldsymbol{D}_i$ design. When Eq. (24) is examined, it can be seen that a $\tan(\theta)$ is used in the problem formulation, which at launch approaches infinity. Taking the tan(90) would cause a large-control-for-large-state problem in a standard optimal control technique. However, the $\boldsymbol{D}_i$ terms in θ–D were designed to handle such a problem through the small number $\varepsilon_i(t) = 1 - k_i e^{-a_i t}$. The role of ε_i has been discussed in Sec. II. The Euler pitch angle θ drops below 85 deg approximately 15 s into the flight, and after that point, the $\tan(\theta)$ is not a problem. With this in mind, a_1 and a_2 values were selected as 0.01, which ensures they cover the appropriate area of the flight and die out after the Euler pitch angle becomes reasonable. As mentioned in Sec. II, a systematic approach to determine the $\boldsymbol{D}_i$ terms was developed after this RLV work was completed. This approach is given in the Appendix.

The terms to be defined at the beginning of this controller design are $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{A}_0$, $\boldsymbol{g}_0$, $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and θ. With $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{A}_0$, and $\boldsymbol{g}_0$ selected, the algebraic Riccati equation (11) can be solved to determine the $\boldsymbol{T}_0$ matrix, which remains constant throughout the solution process. Once these values are defined, a closed-form suboptimal controller can be obtained by applying the θ–D algorithm in Eqs. (12–14). Online implementation of the resulting off-line controller requires little computational power.

V. Simulation Results

The controller developed here was implemented in a high-fidelity simulation of the X-33 RLV. The X-33 was a half-scale technology demonstrator of a proposed RLV, the Venture Star, that was under development by NASA and Lockheed Martin Corp. The X-33 has 12 control effectors that it uses to control its orientation in flight. They are as follows: right and left flaps, right and left inboard elevons, right and left outboard elevons, right and left rudders, and four differentially throttled quadrants on the linear aerospike engine. Electromechanical actuators move these effectors as required by the actuator commands from the control allocator. The X-33 vehicle can be seen in Fig. 1. Because of the planned development of the X-33, NASA MSFC developed a high-fidelity simulation (MAVERIC) for the vehicle that takes into account fuel slosh, actuator nonlinearities, winds, and other vehicle parameters that were obtained from actual hardware testing.

Fig. 1 X-33 RLV.

NASA MSFC uses MAVERIC to test advanced control techniques. Testing is done to compare emerging control methods on a level playing field to see how well they perform against each other. The testing suite set forth by MSFC is a rigorous set, with a total of 46 individual test cases. These test cases examine a wide range of vehicle conditions, including a nominal launch, an engine loss, locked control effectors, varying wind cases, and Monte Carlo dispersions. Such a wide range is tested to determine the robustness of the controller to varying flight and vehicle conditions. In each of these test cases, there are more than 150 parameters examined. The developed controller was implemented and tested in the MAVERIC X-33 simulation.

The designed θ–D controller was first implemented in a MATLAB® model. A MATLAB model was employed so that some tuning runs could be done before implementing the controller in the 6-DOF simulator. Only the rotational dynamics in Eq. (21) were used in the MATLAB model, to examine the effect of variations on $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{A}_0$, $\boldsymbol{g}_0$, $\boldsymbol{D}_1$, and $\boldsymbol{D}_2$. When this model tracked well, the controller with the tuned values was imported into the 6-DOF simulator and then run. Tuning the controller in this way introduced the possibility that the translational motion, which was not accounted for in the MATLAB model, could cause problems when running in the 6-DOF simulator. However, the results presented here show that the controller was able to track well with the translational motion included. The major objectives of the tests with the high-fidelity model were to investigate 1) the θ–D controller performance in a highly nonlinear environment, 2) the θ–D performance as compared to the fine-tuned baseline X-33 PID controller, and 3) how robust the θ–D controller is to changes in the system model. The design methodology presented in this paper outlines the development of a θ–D controller. The control allocator was developed and provided by MSFC in the MAVERIC simulator, which is the case for a majority of the control methods they have tested. The objective of this work was to compare the results of a θ–D controller with a fine-tuned PID controller, not to develop a new control allocator.

The $\boldsymbol{Q}, \boldsymbol{A}_0, \boldsymbol{T}_0, \boldsymbol{D}_1$, and $\boldsymbol{D}_2$ were the same for all runs in the 6-DOF simulator and were chosen to be

$$\boldsymbol{A}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0.0002 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 11.4301 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11.4737 & 0 & 0 & 0 \end{bmatrix} \tag{36}$$

$$\boldsymbol{Q} = \begin{bmatrix} 150 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 150 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 150 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100000 \end{bmatrix} \tag{37}$$

$$\boldsymbol{D}_i = e^{-0.01t}\left[\boldsymbol{T}_{i-1}\boldsymbol{A}(\boldsymbol{x})/\theta - \boldsymbol{A}^T(\boldsymbol{x})\boldsymbol{T}_{i-1}/\theta\right], \qquad i = 1, 2 \tag{38}$$

The $\boldsymbol{R}$ matrix, as explained in Sec. IV, is $\boldsymbol{R} \in R^{3\times 3}$. The $\boldsymbol{R}$ matrix is chosen as a function of the moments of inertia, I_{xx}, I_{yy}, I_{zz}, and I_{xz}, of the RLV to better track the true vehicle mass (as explained in Sec. IV). The symmetric matrix $\boldsymbol{T}_0$ was solved using the $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{g}_0$, and $\boldsymbol{A}_0$ in Eq. (11) and is

$$\boldsymbol{T}_0 = \begin{bmatrix} 505 & 0 & -8.68 & 123 & 0 & 194 & 350 & 0 & -228 \\ 0 & 1344 & 0 & 0 & 387 & 0 & 0 & 1020 & 0 \\ -8.68 & 0 & 3909 & -12.9 & 0 & 1210 & -27.2 & 0 & 935 \\ 122 & 0 & -12.9 & 287 & 0 & 232 & 411 & 0 & -343 \\ 0 & 387 & 0 & 0 & 2634 & -0.33 & 0 & 3470 & 0 \\ 194 & 0 & 1210 & 232 & -0.33 & 85788 & 581 & 0 & 31508 \\ 350 & 0 & -27.2 & 411 & 0 & 581 & 1059 & 0 & -717 \\ 0 & 1020 & 0 & 0 & 3470 & 0 & 0 & 8754 & 0 \\ -228 & 0 & 935 & -343 & 0 & 31508 & -717 & 0 & 24263 \end{bmatrix} \tag{39}$$

These values of Eqs. (36–39) were not chosen arbitrarily. A range of $\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{A}_0, \boldsymbol{g}_0, \boldsymbol{D}_1$, and $\boldsymbol{D}_2$ terms were run in the MATLAB model. After tuning these parameters in the MATLAB model until an acceptable level of performance was achieved, the best values were implemented in the MAVERIC simulator.

The ascent phase of a RLV follows a preset path from launch to main engine cutoff. To follow this path, the simulator issues a set of commanded Euler angles that are scheduled in reference to the magnitude of relative velocity of the vehicle. Although the ascent problem does not require any body rate tracking, the formulation in this study has desired body rates of zero. (They are small in any typical run.) When the weights are suitably adjusted, high for Euler angles and low for body rates, in the optimal control formulation, the controller is able to achieve excellent tracking of Euler angles. Note that in the weights in Eq. (37) the middle three diagonal terms correspond to the body rates and the last three diagonal terms correspond to the Euler angle tracking errors. It can easily be seen that the weighting on the Euler angle tracking errors is much larger than the weighting on the body rates (five to seven orders of magnitude). As mentioned earlier, there are 46 separate launch profiles. The results presented in this paper will focus on three specific tests that are representative of the overall results obtained: a nominal launch, a failed aerosurface, and an engine loss.

A nominal run in the simulation uses a smoothed reference wind and assumes all control surfaces are functioning properly throughout the entire flight profile. The nominal ascent case was run in the MAVERIC simulator. Euler angle tracking for the θ–D and PID controllers in the nominal ascent case are shown in Figs. 2–4. The body rates for the θ–D controller in the nominal ascent case are in shown in Fig. 5. The commanded torques for the θ–D and PID controllers in a nominal ascent are shown in Figs. 6–8. When Figs. 2–4 are examined, it can be seen that the Euler angle tracking errors are within 1–2 deg throughout most of the launch phase. The one exception to this occurs in the pitch plane at approximately 150 s. At this time, there is a scheduled thrust reduction in the engines that the controller has no knowledge of beforehand. Even with this unknown thrust reduction, the error does not grow larger than 4 deg, and it is the same magnitude as the error in the baseline controller.

Figure 5 shows that the body rates are small for the entire ascent phase with the largest absolute value being 3 deg/s and most of the values staying in the range of 1 deg/s. Examining Figs. 6–8 shows that the θ–D allocated torques are comparable to the PID controller torques. The one exception occurs again at 150 s in the pitch plane and occurs because the error is slightly larger there. This spike is within the acceptable bounds. Also of note is the yaw plane torque. Figure 8 shows that the PID yaw torque exhibits oscillatory behavior from 100 to 200 s of flight, whereas the θ–D yaw torque is smoother.

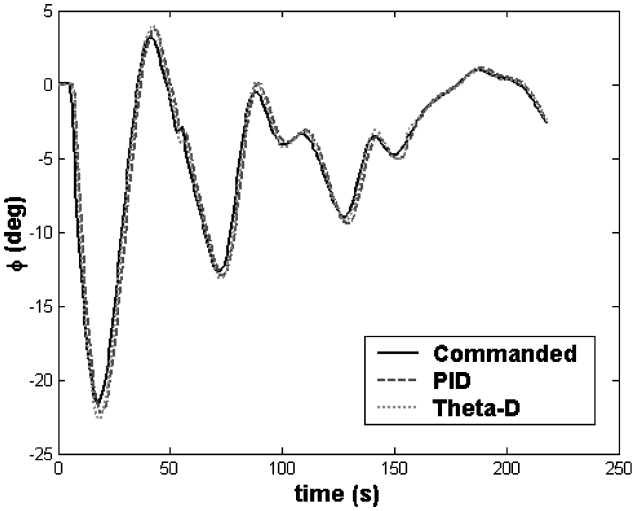

Fig. 2 Nominal roll angle tracking.

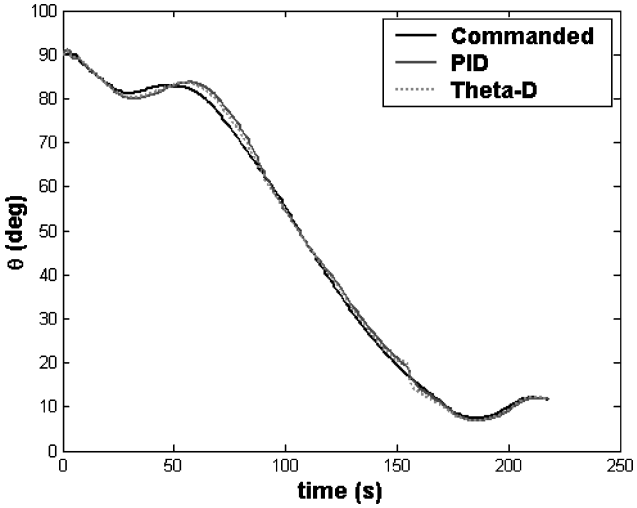

Fig. 3 Nominal pitch angle tracking.

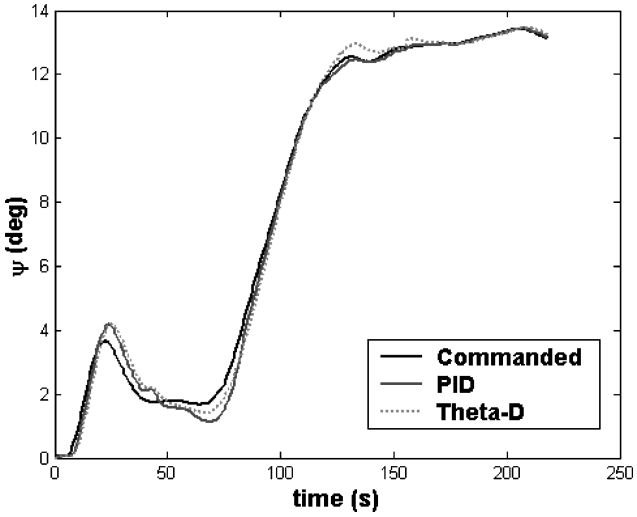

Fig. 4 Nominal yaw angle tracking.

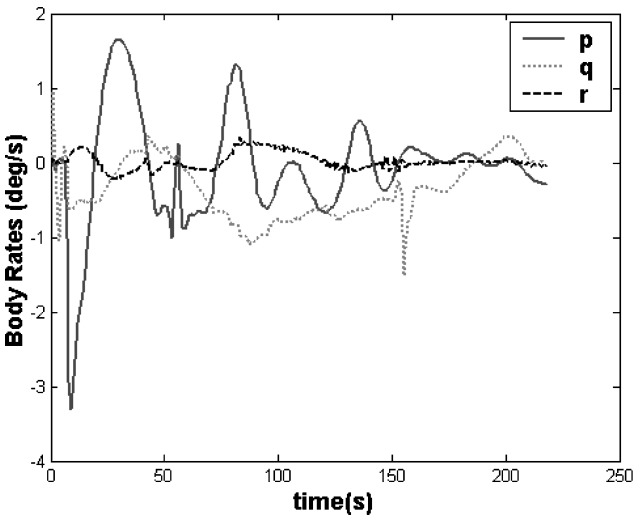

Fig. 5 Nominal body rates.

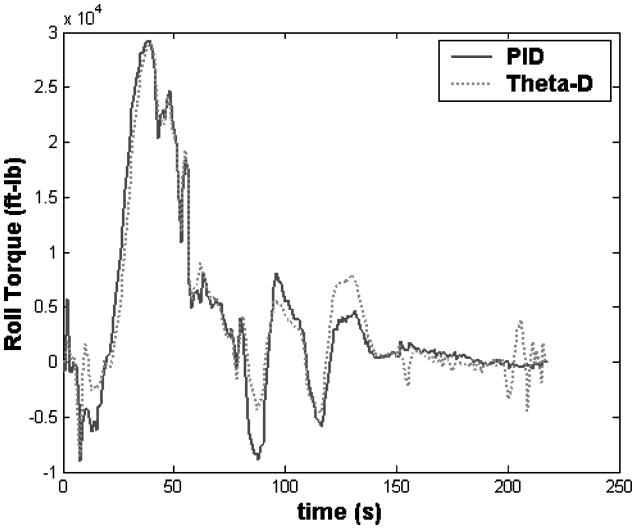

Fig. 6 Nominal roll plane torque.

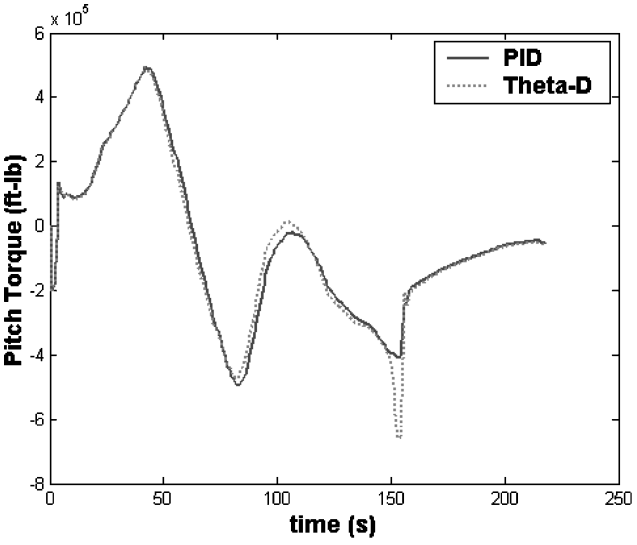

Fig. 7 Nominal pitch plane torque.

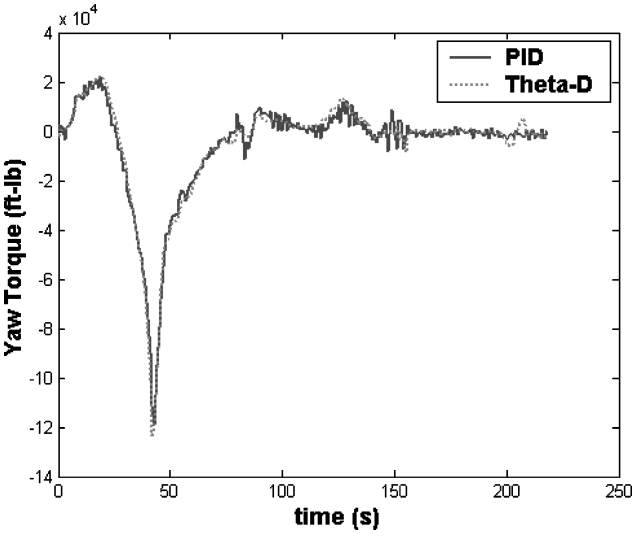

Fig. 8 Nominal yaw plane torque.

One of the multiple possible aerosurface failures is for an elevon to lock at a particular angle during the flight. An aerosurface failure has the potential to cause problems because the elevons are used to control the pitch and roll motion of the RLV. A right inboard elevon locked at 10 deg of deflection was simulated at 50 s into the flight. The right inboard elevon fails to this position for 30 s and then reverts to normal operation. Euler angle tracking for the θ–D and PID controllers in the aerosurface failure case are shown in Figs. 9–11. The commanded torques for the θ–D and PID controllers in the aerosurface failure case are presented in Figs. 12–14. The same trends in the Euler angle error can be seen in the flap failure run as in the nominal run. Euler tracking errors do not grow larger than 2 deg throughout the launch. When the pitch plane is examined at 150 s, the Euler angle error and the torques behave in a similar fashion as in the nominal run.

The X33 is capable of reconfiguring its engines in the event that one of them fails.[17] This is done by redirecting the propellant flow from the functioning engine to both the functioning and the failed engine. Interisolation valves between the two engines are opened to allow the turbopumps from the working engine to supply fuel and oxidizer to both engines. Such an occurrence is referred to as a power pack out (PPO) and allows both engines to provide thrust vector control at half of the nominal values. A PPO was simulated at 105 s in the flight. The Euler angle histories with the θ–D and PID controllers in the PPO case are presented in Figs. 15–17. The commanded torques for the θ–D and PID controllers in the PPO case are given in Figs. 18–20. When Figs. 15–17 are examined, it

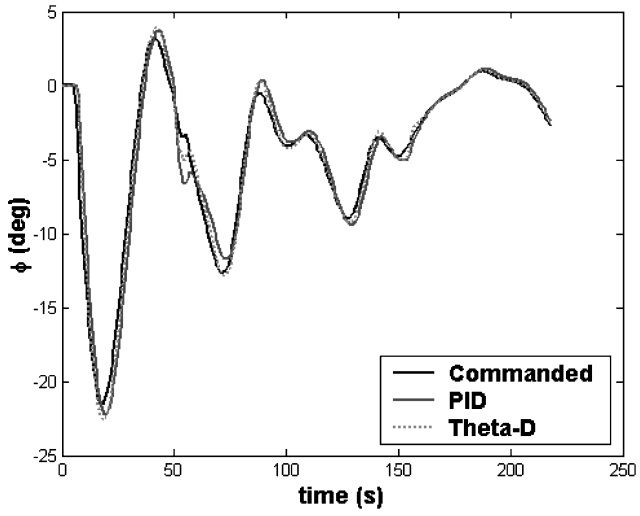

Fig. 9 Flap failure roll angle tracking.

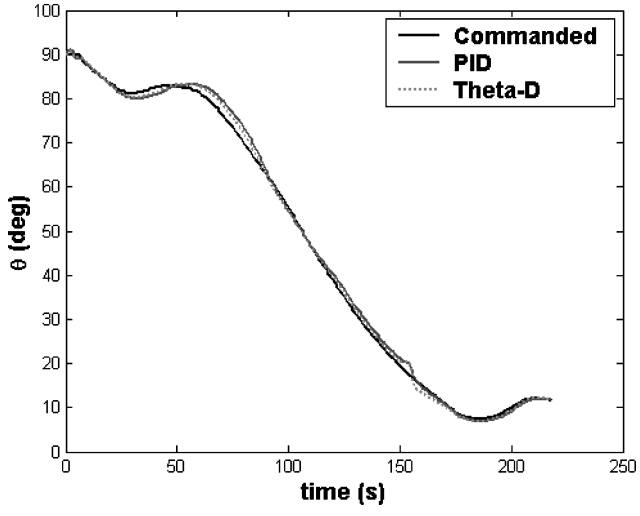

Fig. 10 Flap failure pitch angle tracking.

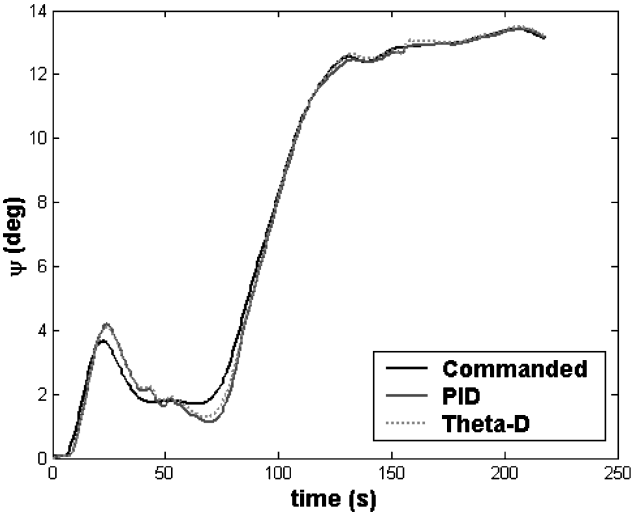

Fig. 11 Flap failure yaw angle tracking.

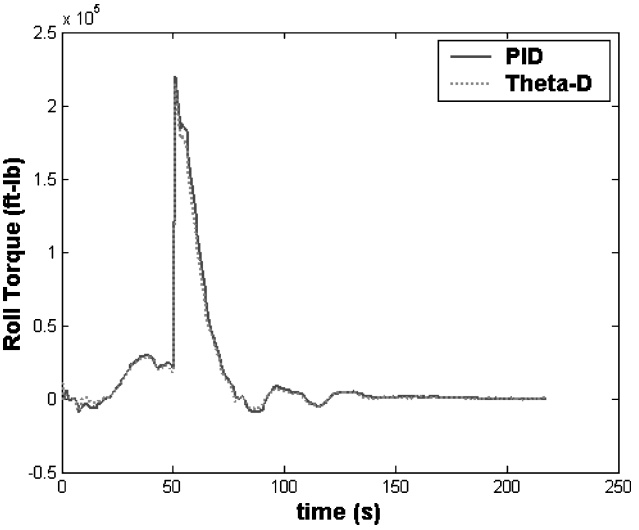

Fig. 12 Flap failure roll plane torque.

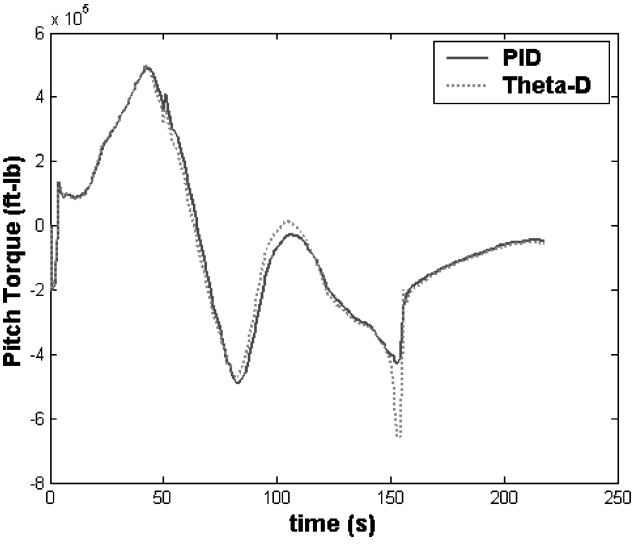

Fig. 13 Flap failure pitch plane torque.

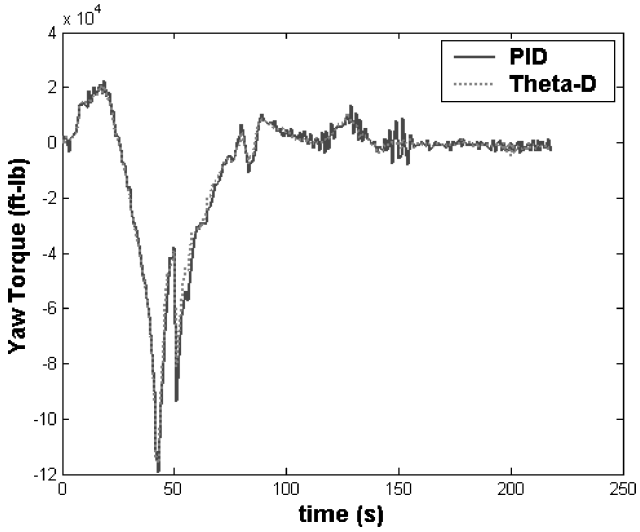

Fig. 14 Flap failure yaw plane torque

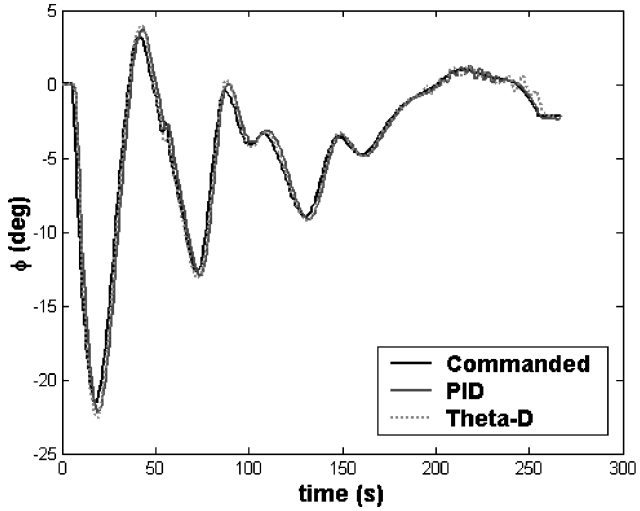

Fig. 15 PPO roll angle tracking.

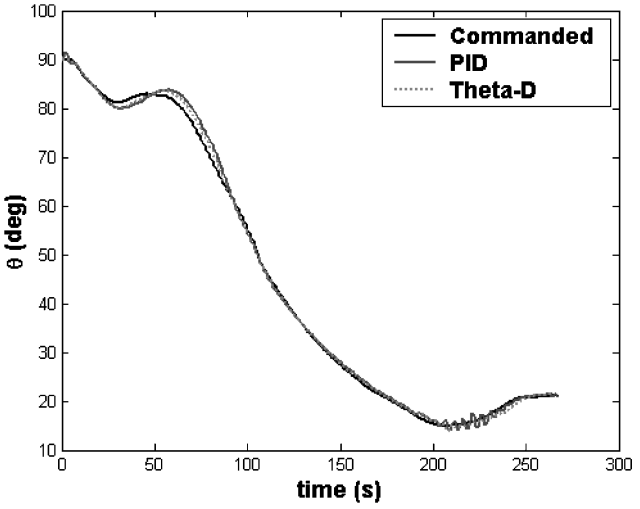

Fig. 16 PPO pitch angle tracking.

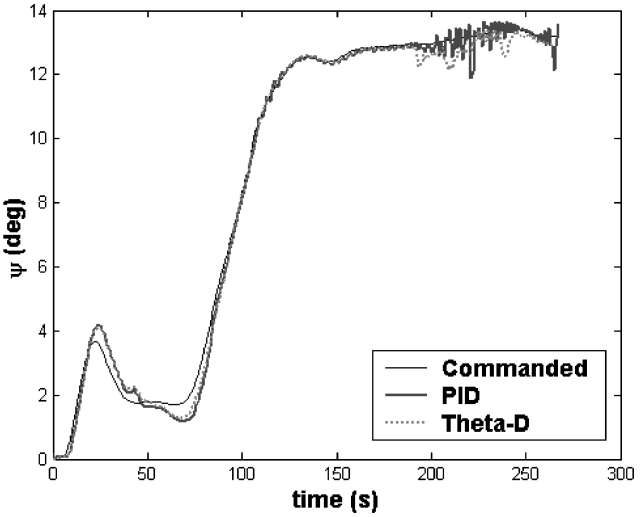

Fig. 17 PPO yaw angle tracking.

is seen that the PPO Euler tracking errors have the same trends as the nominal run tracking errors with the largest tracking error being 2 deg. The spike at 150 s does not occur because there is no scheduled thrust reduction because the vehicle already has a reduced thrust due to the PPO. Torque histories are also affected by this engine loss, as is expected. Pitch and yaw torques have the same history as the nominal case; however, the roll torque has different trends during the last 200 s. Note also that the flight time has increased from approximately 200 s to approximately 275 s because of the reduction in engine power. Of particular interest is the oscillation in the torques in Figs. 18–20. The θ–D controller generated a slight increase in oscillation in the roll plane torque. However, the PID controller has significantly larger oscillations in the pitch and yaw torques when compared to the θ–D controller.

The results from these three test cases show that the Euler angle tracking and torques were comparable to the PID control law used by the X33. When the results are examined, it is easily seen that the θ–D controller tracked as well as the PID controller and better in some cases. The scoring of the θ–D controller throughout all 46 cases shows that this new controller outperforms the PID control algorithm.[18] However, the major difference in the synthesis of the PID controller and the θ–D controller is that the development time for the former was about four years and the latter was about nine months. Also, the θ–D controller uses one set of parameters for the entire range of flight conditions, whereas the PID controller

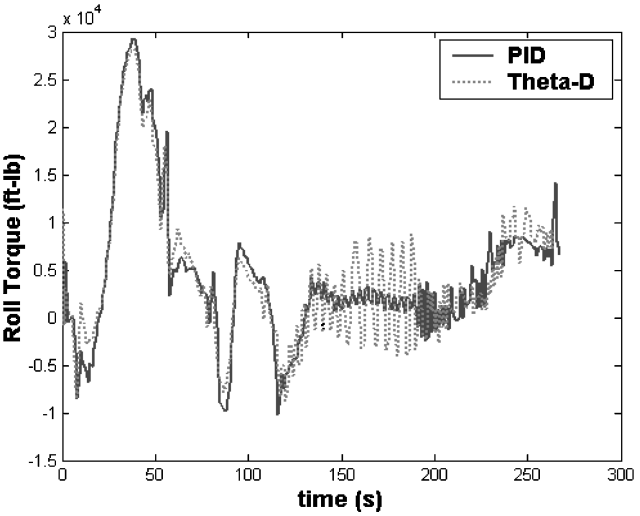

Fig. 18 PPO roll plane torque.

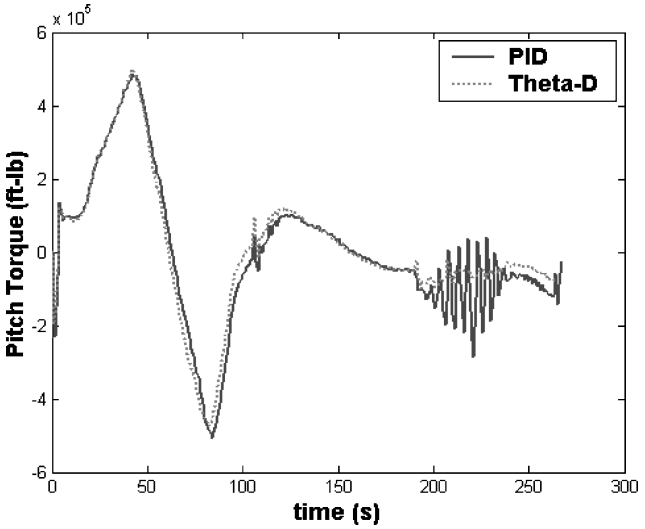

Fig. 19 PPO pitch plane torque.

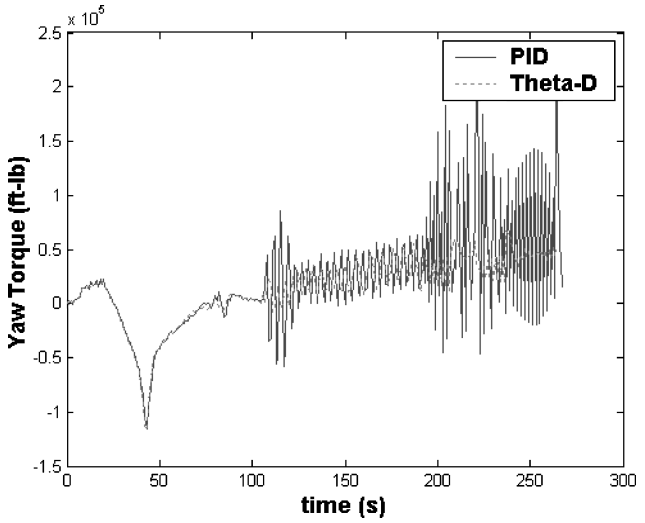

Fig. 20 PPO yaw plane torque.

requires 7800 gain values to cover the range of flight conditions tested. Although both controllers used the same control allocator developed by MSFC, the control allocator is designed with only the prior knowledge of PPO failures. The θ–D controller has proven throughout the simulations that it is capable of handling many disturbances and uncertainties that are unaccounted for in its design.

VI. Conclusions

A suboptimal θ–D controller has been designed for the RLV described in this paper. The θ–D controller was implemented in a 6-DOF high-fidelity simulation, and the results show the controller provides robust and accurate tracking of the RLV Euler angles. The θ–D controller does not require online solution of the algebraic Riccati equation, as some other optimal control techniques do. Also the θ–D controller design is portable inasmuch as there were very few vehicle-specific parameters used in the design or the implementation of the algorithm. This ability to use one set of gains throughout all test cases is an improvement over the PID controller that requires a set of scheduled gains for each different case it will encounter. The reduction in time and tuning can reduce the cost associated with orbit changes, vehicle failure, or unmodeled dynamics of RLVs. The θ–D approximation technique shows promise in the application of control to the RLV ascent phase, and it may be used for other flight phases as well with minor changes in the design parameters.

Appendix: Systematic Determination of k_i and a_i Parameters in D_i Matrices

A systematic way to select the values for k_i and a_i is based that the θ–D approach gives an approximate closed-form solution to the SDRE,

$$\boldsymbol{F}^T(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x}) + \boldsymbol{P}(\boldsymbol{x})\boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{P}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x}) + \boldsymbol{Q} = \boldsymbol{0} \quad \text{(A1)}$$

where $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{A}_0 + \boldsymbol{A}(\boldsymbol{x})$.

To see this, assume that

$$\frac{\partial V}{\partial \boldsymbol{x}} = \boldsymbol{P}(\boldsymbol{x})\boldsymbol{x} \quad \text{(A2)}$$

and $\boldsymbol{P}(\boldsymbol{x})$ is symmetric. When Eq. (A2) into is substituted the HJB equation,

$$\frac{\partial V^T}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) - \frac{1}{2}\frac{\partial V^T}{\partial \boldsymbol{x}}\boldsymbol{B}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{B}^T(\boldsymbol{x})\frac{\partial V}{\partial \boldsymbol{x}} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} = 0 \quad \text{(A3)}$$

and nonlinear $\boldsymbol{f}(\boldsymbol{x})$ is written in a linearlike structure, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x})\boldsymbol{x}$ leads to the SDRE (A1). Compared with the SDRE (A1), the θ–D method solves a perturbed HJB equation and assumes that

$$\frac{\partial V}{\partial \boldsymbol{x}} = \sum_{i=0}^{\infty} \boldsymbol{T}_i(\boldsymbol{x}, \theta)\theta^i \boldsymbol{x} \quad \text{(A4)}$$

As can be seen, the θ–D method is similar to the SDRE approach in the sense that both bring the nonlinear equation into a linearlike structure $f(x) = F(x)x$ and solve the optimal control problem. The former gives an approximate closed-form solution, whereas the latter solves the algebraic Riccati equation at each state. It can be predicted that the θ–D solution is close to the SDRE solution, that is,

$$\boldsymbol{P}(\theta - D) \approx \boldsymbol{P}(\text{SDRE})$$

where

$$\boldsymbol{P}(\theta - D) = \sum_{i=0}^{\infty} \boldsymbol{T}_i(\boldsymbol{x}, \theta)\theta^i$$

is obtained from solving Eqs. (11–14) in the paper off-line and $\boldsymbol{P}$(SDRE) is obtained by solving Eq. (A1) online.

An efficient procedure for finding the (k_i, a_i) was determined as follows: The SDRE controller is used to generate a state trajectory and then the maximum singular value of $\boldsymbol{P}$(SDRE), that is, $\sigma_{\max}[\boldsymbol{P}(\text{SDRE})]$, is computed at each state point. Similarly, the (k_i, a_i) parameters determine $\boldsymbol{P}(\theta$–$D)$ and its associated $\sigma_{\max}[\boldsymbol{P}(\theta$–$D)]$. Curve fits are then applied to $\sigma_{\max}[\boldsymbol{P}(\text{SDRE})]$ and $\sigma_{\max}[\boldsymbol{P}(\theta$–$D)]$, and the (k_i, a_i) are selected to minimize the difference between these singular value histories in a least-square sense. The (k_i, a_i) can all be determined in one least-square run off-line.

Acknowledgments

Support for this study from the U.S. Naval Surface Warfare Center, Dahlgren, Virginia, and NASA Marshall Spaceflight Center, Huntsville, Alabama, is gratefully acknowledged.

References

[1]Hanson, J. M., "A Plan For Advanced Guidance and Control Technology for 2nd Generation Reusable Launch Vehicles," AIAA Paper 2002-4557, Aug. 2002.

[2]Hanson, J. M., and Jones, R. E., "Advanced Guidance and Control Methods For Reusable Launch Vehicles: Test Results," AIAA Paper 2002-4561, Aug. 2002.

[3]Shtessel, Y., Zhu, J., and Daniels, D., "Reusable Launch Vehicle Attitude Control Using a Time-Varying Sliding Mode Control Technique," AIAA Paper 2002-4779, Aug. 2002.

[4]Johnson, E. N., and Calise, A. J., "Limited Authority Adaptive Flight Control for Reusable Launch Vehicles," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 6, 2003, pp. 906–913.

[5]Shtessel, Y., Hall, C., and Jackson, M., "Reusable Launch Vehicle Control in Multiple-Timescale Sliding Modes," *Journal of Guidance, Control and Dynamics*, Vol. 23, No. 6, 2000, pp. 1013–1020.

[6]Shtessel, Y., Stott, J., and Zhu, J., "Time-Varying Sliding Mode Control with Sliding Mode Observer for Reusable Launch Vehicle," AIAA Paper 2003-5362, Aug. 2003.

[7]Zhu, J., Banker, B. D., and Hall, C. E., "X-33 Ascent Flight Controller Design by Trajectory Linearization—A Singular Perturbational Approach," AIAA Paper 2000-4159, Aug. 2000.

[8]Doman, D. B., and Ngo, A. D., "Dynamic Inversion-Based Adaptive/Reconfigurable Control of the X-33 on Ascent," *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 2, 2002, pp. 275–284.

[9]Hodel, A. S., and Callahan, R., "Autonomous Reconfigurable Control Allocation (ARCA) for Reusable Launch Vehicles," AIAA Paper 2002-4777, Aug. 2002.

[10]Safonov, M. G., and Athans, M., "Gain and Phase Margins of Multiloop LQG Regulators," *IEEE Transactions on Automatic Control*, Vol. 22, No. 2, 1977, pp. 173–179.

[11]Huang, Z., and Balakrishnan, S. N., "Robust Adaptive Critic Based Neuro Controllers for Helicopters with Unmodeled Uncertainties," AIAA Paper 2001-4258, Aug. 2001.

[12]Cloutier, J. R., D'Souza, C. N., and Mracek, C. P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State-Dependent Riccati Equation Technique," *Proceedings of 1st International Conference on Nonlinear Problems in Aviation and Aerospace*, Vol. 1, AIAA, Reston, VA, 1996, pp. 117–123.

[13]Xin, M., and Balakrishnan, S .N., "A New Method for Suboptimal Control of A Class of Nonlinear Systems," *Proceedings of IEEE Conference on Decision and Control*, IEEE Control System Society, Vol. 3, IEEE Publications, Piscataway, NJ, 2002, pp. 2756–2761.

[14]Xin, M., and Balakrishnan, S. N., "Missile Longitudinal Autopilot Design Using a New Suboptimal Nonlinear Control Method," *IEE Proceedings on Control Theory and Applications*, Vol. 150, No. 6, 2003, pp. 577–584.

[15]Xin, M., Balakrishnan, S. N., Stansbery, D. T., and Ohlmeyer, E. J., "Nonlinear Missile Autopilot Design with Theta-D Technique," *Journal of Guidance, Control and Dynamics*, Vol. 27, No. 3, 2004, pp. 406–417.

[16]Bryson, A. E., and Ho, Y., *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 128–148.

[17]Hall, C. E., and Panossian, H. V., "X-33 Attitude Control Using the XRS-2200 Linear Aerospike Engine," AIAA Paper 99-2936, June 1999.

[18]Hanson, J. M., Jones, R. E., Hall, C. E., and Mulqueen, J. A., "Advanced Guidance and Control for Hypersonics and Space Access," JANNAF Interagency Propulsion Committee Joint Subcommittee Meeting, Dec. 2003.