AIAA Guidance, Navigation, and Control Conference and Exhibit
1111 - 14 August 2003, Austin, Texas

AIAA 2003-5634

EXTENSION OF A PSEUDOSPECTRAL LEGENDRE METHOD TO NON-SEQUENTIAL MULTIPLE-PHASE OPTIMAL CONTROL PROBLEMS

Anil V. Rao*
The Charles Stark Draper Laboratory, Inc.
Cambridge, MA 02139-3563

Abstract

A pseudospectral Legendre method for discretizing continuous-time optimal control problems is extended to non-sequential multiple-phase optimal control problems. In the method developed in this paper, each phase of the problem is treated independently. The phases are then linked together via continuity conditions on the independent variable, the state, and the control. In this manner, it is possible to decouple the problem into distinct segments and to specify a different mathematical form and dimension for the state, control, and constraints in each phase. Moreover, because the phases need not be sequential, it is possible to optimize trajectories for problems involving multiple systems that may share information. The multiple-phase optimal control problem is discretized in each phase using a previously developed single phase pseudospectral Legendre method. This discretization leads to a nonlinear programming problem (NLP) that has a well-defined structure. The structure of the NLP is described in detail and a particular software implementation is discussed. The software is demonstrated on an example that exhibits the key features of the multiple-phase pseudospectral discretization. The method developed here is an efficient approach for solving general non-sequential multiple-phase optimal control problems numerically.

INTRODUCTION

Numerical methods for solving optimal control problems fall into two general categories: indirect methods and direct methods. An excellent survey of various numerical methods can be found in Ref. 5. In an indirect method, the first-order optimality conditions are derived using the minimum principle of Pontryagin.[1] These necessary conditions lead to a Hamiltonian boundary-value problem (HBVP) which is then solved to determine candidate optimal trajectories called extremal trajectories. In a direct method, the optimal control problem is discretized at specified time points called nodes. Common discretization methods include trapezoidal, Hermite-Simpson, and Runge-Kutta.[6] Any of these discretization methods leads to a nonlinear programming problem (NLP) which is then solved using an appropriate optimization method such as those described in Ref. 3 and Ref. 12.

A key benefit of indirect methods is that, by solving the HBVP, the co-state of the optimally controlled system is obtained. The co-state offers the opportunity to gain a great deal of insight about the underlying structure of the optimal solution. However, a significant drawback to indirect methods is that many HBVPs are *hyper-sensitive*[15,16] and thus suffer from ill-conditioning due to extreme sensitivity of the dynamics of the Hamiltonian system in a neighborhood of the optimal solution.[15,16] Because of this sensitivity, obtaining an extremal trajectory often requires an unreasonably good initial guess. Unfortunately, it is often not possible to find such a high-quality initial

*Senior Member of the Technical Staff
Guidance and Navigation Division
Address for Correspondence: 555 Technology Square, Mail Stop 70, Cambridge, MA 02139-3563.
E-mail: arao@draper.com.

guess. Consequently, it is often difficult to obtain numerical solutions to optimal control problems using indirect methods.

In recent years, direct methods have risen to prominence. Direct methods have an advantage over indirect methods in that it is possible to solve extremely complex problems with relatively poor initial guesses. The reason that direct methods are superior to indirect methods is that finding a solution to the NLP is often significantly easier than finding a solution to the corresponding HBVP. Well-known software packages employing direct methods include *Optimal Trajectories by Implicit Simulation* (OTIS),[23] *Sparse Optimal Control Software* (SOCS),[4] *Graphical Environment for Simulation and Optimization* (GESOP),[24] and *Direct and Indirect Dynamic Optimization* (DIDO).[21]

A recently developed class of direct methods for solving optimal control problems is the class of so-called *pseudospectral methods*.[9,10,14,18,21] In a pseudospectral method, the state and control are approximated using a basis of global orthogonal polynomials. Using this approximation and a proper choice for the nodes, an efficient discretization is obtained from which an accurate approximation to the optimal trajectory and optimal control of the continuous-time optimal control problem is obtained. It is also important to note that, due to its structure, a pseudospectral discretization is significantly easier to implement than many other direct methods.

A particular pseudospectral method that has shown promise in computational optimal control is the so-called *Pseudospectral Legendre Method*.[10,19] In the pseudospectral Legendre method, the state and control are approximated by globally interpolating Lagrange polynomials and the nodes are the *Legendre-Gauss-Lobatto* points. The pseudospectral Legendre method has been shown to produce accurate solutions on relatively simple problems.[10,18,19] Furthermore, the co-state estimation procedure developed in[19] has shown to produce accurate co-states on simple problems.[19]

While the pseudospectral Legendre method of Ref. 10 and Ref. 19 has been shown to be extremely viable on simple problems, the method presented in Ref. 10 and Ref. 19 has three key related limitations. First, the method of Ref. 10 and Ref. 19 does not allow for interior point constraints. Second, commensurate with the first limitation, the method of Ref. 10 and Ref. 19 does not allow for changes in the mathematical form for the dynamics or constraints within the discretization. Third, commensurate with the first two limitations, the method of Ref. 10 and Ref. 19 does not permit changes in the dimensions of the state, control, or constraints within the discretization.

The three aforementioned limitations of the pseudospectral Legendre method of Ref. 10 and Ref. 19 are quite severe for many reasons. First, many optimal control problems have interior point constraints. Second, from a computational standpoint it may be desirable to divide an optimal control problem into distinct *phases* (i.e. distinct non-overlapping segments of the complete trajectory) where a unique mathematical description is used within each phase. Finally, in addition to the need to decompose the problem into phases, it may be necessary to specify a different dimension for the variables and constraints in each phase of the problem (e.g. one phase of a problem may have mass as a state while another phase may have a constant mass). An excellent example of the need to decompose a problem into phases is an aeroassisted orbital transfer where a vehicle has atmospheric and exo-atmospheric segments that need to be modeled differently; the reader is referred to Ref. 17 for details about such a problem. Because of its form, the method described in Ref. 10 and Ref. 19 cannot be implemented on such problems. Consequently, in order to increase the range of applicability of the pseudospectral Legendre method, it is useful to develop an extension of the method of Ref. 10 and Ref. 19 to the class of multiple-phase optimal control problems.

Two approaches have been previously developed for solving multiple-segment optimal control problems using pseudospectral methods. The first approach is called *spectral patching*.[18] In spectral patching, the optimal control problem is split into two or more segments and the pseudospectral Legendre discretization is applied on each segment. The segments are then patched together using continuity conditions on the state and control. While this approach has proven useful, the spectral patching procedure developed in Ref. 18 is restricted to problems where the dynamic model is the exactly same on each segment and thus is not applicable to problems with different mathematical descriptions in each segment. Furthermore, the procedure of Ref. 18 does not allow for discontinuities in the state at the boundary between two segments and thus is not applicable to problems with different mathematical descriptions in each segment.

The second approach for solving multiple-segment problems via pseudospectral methods is applied to so-called *nonsmooth optimal control problems* and has been developed in Ref. 20. As with spectral patching, in the nonsmooth method of Ref. 20 the original optimal control problem is again split into two or more segments and the pseudospectral discretization is applied to each segment. However, unlike spectral patching, this second approach allows for discontinu-

ities in the state and control at the interface between two segments (i.e. the method allows for *nonsmoothness*). While the method of Ref. 20 is more general than the method described in Ref. 18, it still has the following limitations. First, the mathematical form of the interface conditions does not allow for transformations in either the state or the control. Second, the method presented in Ref. 20 does not allow for changes in the dimension of the state between segments. Third, because the formulation developed in Ref. 20 does not decouple the problem into distinct phases, it makes it difficult to conceptualize the problem in terms of parts that are each distinct from one another. Fourth, the formulation in Ref. 20 does not address the issue of solving non-sequential multiple-phase optimal control problems.

In this paper, the pseudospectral Legendre method of Ref. 10 is extended to general non-sequential multiple-phase optimal control problems. In the extension developed here, the models chosen for the problem may be different in different phases. In particular, the modeling differences in different phases include different descriptions and dimensions for the state and control and, correspondingly, the models used to describe the differential equations, path constraints, and endpoint constraints. Within every phase, the state and control are discretized using the method of Ref. 10. The phases are then linked by enforcing continuity on the independent variable, the state, and the control at the phase boundaries. In general, the conditions that link two phases consist of a set of nonlinear functions and may be used to link non-sequential phases. Moreover, because the phases need not be sequential, the approach developed enables optimization of trajectories for problems involving multiple systems that may share information. Because of its generality, the approach used for phase linking is an adaptation of the approach described in Ref. 6. A key feature of the multiple-phase method developed in this paper is that the resulting optimal control problem is decoupled into distinct segments, each potentially with its own physical and/or mathematical characteristics. The NLP that results from the multiple-phase discretization is then described detail and a particular software implementation of the method is discussed. Finally, the key features of the method are demonstrated on a representative problem. The approach developed here offers a viable approach for efficiently solving general non-sequential multiple-phase optimal control problems.

ONE-PHASE PSEUDOSPECTRAL LEGENDRE METHOD

Consider the following optimal control problem. Minimize the cost functional

$$\mathcal{J} = \varphi[\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f] \int_{t_0}^{t_f} \mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t]dt \quad (1)$$

subject to the dynamic constraints

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \quad (2)$$

the path constraints

$$\boldsymbol{g}_{\min} \leq \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \leq \boldsymbol{g}_{\max} \quad (3)$$

and the initial and terminal constraints

$$\begin{array}{rl} \boldsymbol{h}_{0,\min} & \leq \boldsymbol{h}_0(\boldsymbol{x}(t_0), t_0) \leq \quad \boldsymbol{h}_{0,\max} \\ \boldsymbol{h}_{f,\min} & \leq \boldsymbol{h}_f(\boldsymbol{x}(t_f), t_f) \leq \quad \boldsymbol{h}_{f,\max} \end{array} \quad (4)$$

In Equations (1)-(4) $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the control, $t \in \mathbb{R}$ is the independent variable (generally speaking, time), $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$, $\boldsymbol{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^p$, $\boldsymbol{h}_0 : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{q_0}$, and $\boldsymbol{h}_f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{q_f}$. Consider now the following transformation of the independent variable t:

$$t = [(t_f - t_0)\tau + (t_f + t_0)]/2 \quad (5)$$

where $\tau \in [-1, 1]$. In terms of the variable τ, optimal control problem of Equations (1)-(4) has the following form. Minimize the cost functional

$$\begin{aligned} \mathcal{J} = & \varphi(\boldsymbol{x}(-1), t_0, \boldsymbol{x}(1), t_f) \\ & + \frac{t_f - t_0}{2} \int_{-1}^{1} \mathcal{L}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau, t_0, t_f)d\tau \end{aligned} \quad (6)$$

subject to the dynamic constraints

$$\frac{d\boldsymbol{x}}{d\tau} \equiv \dot{\boldsymbol{x}} = \frac{t_f - t_0}{2} \boldsymbol{f}(x(\tau), u(\tau), \tau, t_0, t_f) \quad (7)$$

the path constraints

$$\boldsymbol{g}_{\min} \leq \boldsymbol{g}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau, t_0, t_f) \leq \boldsymbol{g}_{\max} \quad (8)$$

and the initial and terminal constraints

$$\begin{array}{rl} \boldsymbol{h}_{0,\min} & \leq \boldsymbol{h}_0(\boldsymbol{x}(-1), t_0) \leq \quad \boldsymbol{h}_{0,\max} \\ \boldsymbol{h}_{f,\min} & \leq \boldsymbol{h}_f(\boldsymbol{x}(1), t_f) \leq \quad \boldsymbol{h}_{f,\max} \end{array} \quad (9)$$

The transformed optimal control problem of Equations (6)-(9) is discretized using the pseudospectral Legendre method of Ref. 10. In the method of Ref. 10,

the state and control are approximated using globally interpolating Lagrange polynomials obtained from orthogonal Legendre polynomials.[10,19] Furthermore, the time derivative of the state is approximated using a differentiation matrix derived in a manner consistent with the approximation of the state. Finally, the cost function is computed by approximating the cost functional using the Legendre-Gauss-Lobatto (LGL) quadrature integration rule. In the pseudospectral Legendre method, the discretization points (i.e. nodes) are the LGL points on the transformed time interval $\tau \in [-1, 1]$. For a given choice of N, the LGL points, $\tau_k, k = 0, \ldots, N$, have the property that $\tau_0 = -1$ and $\tau_N = 1$. Then the following approximations are obtained for the state, control, and cost functional. First, the state and control are approximated as

$$\begin{aligned} \boldsymbol{x}(\tau) \approx \boldsymbol{x}^N(\tau) = \sum_{l=0}^{N} \boldsymbol{x}(\tau_l)\phi_l(\tau) \\ \boldsymbol{u}(\tau) \approx \boldsymbol{u}^N(\tau) = \sum_{l=0}^{N} \boldsymbol{u}(\tau_l)\phi_l(\tau) \end{aligned} \tag{10}$$

where

$$\phi_l(\tau) = \frac{1}{N(N+1)L_N(\tau_l)} \frac{(\tau^2-1)\dot{L}_N(\tau)}{\tau - \tau_l}, \quad l = 0, 1, \ldots, N \tag{11}$$

are the Lagrange polynomials of order N on the time interval $\tau \in [\tau_0, \tau_N]$ and $L_N(t)$ is the Legendre polynomial of degree N. It can be shown that[10]

$$\phi_l(\tau_k) = \delta_{lk} = \begin{cases} 1 & \text{if} \quad l = k \\ 0 & \text{if} \quad l \neq k \end{cases}, \quad k, l = 0, 1, \ldots, N \tag{12}$$

Furthermore, from the property of Equation (12) it follows that[10]

$$\begin{aligned} \boldsymbol{x}^N(\tau_k) &= \boldsymbol{x}(\tau_k) \\ \boldsymbol{u}^N(\tau_k) &= \boldsymbol{u}(\tau_k) \end{aligned}, \; k = 0, 1, \ldots, N \tag{13}$$

Differentiating the approximation for $\boldsymbol{x}^N(\tau)$ from Equation (10) and evaluating the result at the LGL points, it can then be shown that[9,10,14]

$$\left[\frac{d\boldsymbol{x}^N}{d\tau}\right]_{\tau=\tau_k} = \sum_{l=0}^{N} D_{kl}\boldsymbol{x}(\tau_l), \; k = 0, 1, \ldots, N \tag{14}$$

where

$$\boldsymbol{D} = [D_{kl}], \; k, l = 0, 1, \ldots, N \tag{15}$$

is the differentiation matrix. The differentiation matrix of Equation 15 has the following property:

$$D_{kl} = \begin{cases} \dfrac{L_N(\tau_k)}{L_N(\tau_l)(\tau_k - \tau_l)} & , \quad k \neq l \\ -\dfrac{N(N+1)}{4} & , \quad k = l = 0 \\ \dfrac{N(N+1)}{4} & , \quad k = l = N \\ 0 & , \quad \text{otherwise} \end{cases} \tag{16}$$

Substituting the result from Equation (14) into Equation (7) and evaluating the function $\boldsymbol{f}$ at the LGL points, the dynamic constraints at the LGL points must satisfy[10]

$$\begin{aligned} \sum_{l=0}^{N} D_{kl}\boldsymbol{x}(\tau_l) - \frac{t_f - t_0}{2}\boldsymbol{f}(\boldsymbol{x}(\tau_k), \boldsymbol{u}(\tau_k), \tau_k, t_0, t_f) \\ = \boldsymbol{0}, \; k = 0, 1, \ldots, N \end{aligned} \tag{17}$$

From this point forth for convenience we make the following substitutions:

$$\begin{aligned} \boldsymbol{x}_k \equiv \boldsymbol{x}(\tau_k) \\ \boldsymbol{u}_k \equiv \boldsymbol{u}(\tau_k) \end{aligned} \tag{18}$$

Then, using the Legendre-Gauss-Lobatto quadrature rule, the cost functional is approximated as

$$\begin{aligned} J \approx J^N = \varphi(\boldsymbol{x}_0, t_0, \boldsymbol{x}_N, t_f) \\ + \frac{t_f - t_0}{2} \sum_{k=0}^{N} \mathcal{L}(\boldsymbol{x}_k, \boldsymbol{u}_k, \tau_k, t_0, t_f) w_k \end{aligned} \tag{19}$$

where $w_k, k = 0, 1, \ldots, N$ are the weights and are given as[10]

$$w_k = \frac{2}{N(N+1)} \frac{1}{L_N^2(t_k)}, \; k = 0, 1, \ldots, N \tag{20}$$

Furthermore, the discretized path constraints are given as

$$\boldsymbol{g}_{\min} \leq \boldsymbol{g}\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \tau_k, t_0, t_f\right) \leq \boldsymbol{g}_{\max}, \; k = 0, 1 \ldots, N \tag{21}$$

Finally, the discretized initial and terminal constraints are given as

$$\begin{aligned} \boldsymbol{h}_{0,\min} \leq \boldsymbol{h}_0(\boldsymbol{x}_0, t_0) \leq \boldsymbol{h}_{0,\max} \\ \boldsymbol{h}_{f,\min} \leq \boldsymbol{h}_f(\boldsymbol{x}_N, t_f) \leq \boldsymbol{h}_{f,\max} \end{aligned} \tag{22}$$

A discrete nonlinear programming problem (NLP) is obtained directly from the approximation to the optimal control problem of Equations (6)-(9) and is stated

as follows. Minimize the cost function of Equation (19) over the variables

$$\begin{array}{c} \boldsymbol{x}_k \in \mathbb{R}^n,\ k = 0, 1, \ldots, N \\ \boldsymbol{u}_k \in \mathbb{R}^m,\ k = 0, 1, \ldots, N \\ t_0 \in \mathbb{R} \\ t_f \in \mathbb{R} \end{array} \tag{23}$$

subject to the nonlinear constraints of Equation (17), Equation (21), and Equation (22). It can be seen that the aforementioned NLP has $(n+m)(N+1)+2$ variables and $(n+p)(N+1)+q_0+q_f$ constraints.

NON-SEQUENTIAL MULTIPLE-PHASE OPTIMAL CONTROL PROBLEMS

Description of a multiple-phase optimal control problem

A multiple-phase optimal control problem is one where the trajectory consists of a collection of *phases*.[6] Simply speaking, a phase is any segment of the complete trajectory. In general, any particular phase of an optimal control problem has a cost functional, a dynamic model, path constraints, and boundary conditions. However, the models used to quantitatively describe the trajectory may be different in different phases of the trajectory. The complete trajectory is then obtained by properly *linking* adjacent phases via linkage conditions.[6] Similarly, the total cost functional is the sum of the cost functionals within each phase. The optimal trajectory is then found by minimizing the total cost functional subject to the constraints within each phase and the linkage constraints connecting adjacent phases. In this section we describe the details of a non-sequential multiple-phase optimal control problem.

Structure of a single phase

Consider an optimal control problem that consists of R distinct phases. For each phase $r \in [1, \ldots, R]$ let

$$\begin{array}{rcl} \boldsymbol{x}^{(r)}(t) & \in & \mathbb{R}^{n^{(r)}} \\ \boldsymbol{u}^{(r)}(t) & \in & \mathbb{R}^{m^{(r)}} \end{array} \tag{24}$$

denote the state and control, respectively. Furthermore, let the independent variable in phase $r \in [1, \ldots, R]$ lie on the interval $t \in \left[t_0^{(r)}, t_f^{(r)}\right]$. Then the cost functional, $\mathcal{J}^{(r)}$, in phase $r \in [1, \ldots, R]$ is given in the form of Equation (1) as

$$\begin{aligned} \mathcal{J}^{(r)} = & \varphi^{(r)}(\boldsymbol{x}^{(r)}(t_0^{(r)}), t_0^{(r)}, \boldsymbol{x}^{(r)}(t_f^{(r)}), t_f^{(r)}) \\ & + \int_{t_0^{(r)}}^{t_f^{(r)}} \mathcal{L}^{(r)}(\boldsymbol{x}^{(r)}(t), \boldsymbol{u}^{(r)}(t), t) dt \end{aligned} \tag{25}$$

Similarly, the dynamic constraints in phase $r \in [1, \ldots, R]$ are given as

$$\frac{d\boldsymbol{x}^{(r)}}{dt} = \boldsymbol{f}^{(r)}(\boldsymbol{x}^{(r)}(t), \boldsymbol{u}^{(r)}(t), t) \tag{26}$$

where

$$\boldsymbol{f}^{(r)} : \mathbb{R}^{n^{(r)}} \times \mathbb{R}^{m^{(r)}} \times \mathbb{R} \to \mathbb{R}^{n^{(r)}} \tag{27}$$

Furthermore, the path constraints in phase $r \in [1, \ldots, R]$ are given as

$$\boldsymbol{g}_{\min}^{(r)} \leq \boldsymbol{g}^{(r)}(\boldsymbol{x}^{(r)}(t), \boldsymbol{u}^{(r)}(t), t) \leq \boldsymbol{g}_{\max}^{(r)} \tag{28}$$

where

$$\boldsymbol{g}^{(r)} : \mathbb{R}^{n^{(r)}} \times \mathbb{R}^{m^{(r)}} \times \mathbb{R} \to \mathbb{R}^{p^{(r)}} \tag{29}$$

Finally, the initial and terminal constraints in phase $r \in [1, \ldots, R]$ are given as

$$\begin{array}{ll} \boldsymbol{h}_{0,\min}^{(r)} & \leq \boldsymbol{h}_0^{(r)}(\boldsymbol{x}^{(r)}(t_0), t_0^{(r)}) \leq \boldsymbol{h}_{0,\max}^{(r)} \\ \boldsymbol{h}_{f,\min}^{(r)} & \leq \boldsymbol{h}_f^{(r)}(\boldsymbol{x}^{(r)}(t_f), t_f^{(r)}) \leq \boldsymbol{h}_{f,\max}^{(r)} \end{array} \tag{30}$$

where

$$\begin{array}{c} \boldsymbol{h}_0^{(r)} : \mathbb{R}^{n^{(r)}} \times \mathbb{R} \to \mathbb{R}^{q_0^{(r)}} \\ \boldsymbol{h}_f^{(r)} : \mathbb{R}^{n^{(r)}} \times \mathbb{R} \to \mathbb{R}^{q_f^{(r)}} \end{array} \tag{31}$$

In Equations (25)-(31), the dimensions of the state, control, and all functions change from phase to phase, i.e. the quantities $n^{(r)}$, $m^{(r)}$, $p^{(r)}$, $q_0^{(r)}$, and $q_f^{(r)}$ are in general *different* in each phase.

Linkage conditions between phases

Now consider two phases, $a \in [1, \ldots, R]$ and $b \in [1, \ldots, R]$, of an R-phase optimal control problem (where $a \neq b$). Furthermore, for simplicity in the discussion, assume that $a < b$ (although, strictly speaking, this assumption is not necessary). In general, a and b may not be sequential phases In order for the trajectory in phase a to be connected to the trajectory in phase b, a set of conditions must exist that *link* together phases a and b in meaningful way. Quantitatively, the linkage conditions are a set of equations that describe the way the independent variable, the state, and the control at the terminus of phase a depend upon the independent variable, the state, and the control at the start of phase b. From the description of the previous subsection, the time, state, and

control at the terminus of phase a are given, respectively, as $t_f^{(a)}$, $\boldsymbol{x}^{(a)}(t_f^{(a)})$, and $\boldsymbol{u}^{(a)}(t_f^{(a)})$. Similarly, the time, state, and control at the start of phase b are given, respectively, as $t_0^{(b)}$, $\boldsymbol{x}^{(b)}(t_0^{(b)})$, and $\boldsymbol{u}^{(b)}(t_0^{(b)})$. Then, in order for the independent variable to be continuous between phases a and b, the following linkage condition must be satisfied:

$$t_f^{(a)} - t_0^{(b)} = 0 \tag{32}$$

Second, the state and the control at the terminus of phase a must depend (in some meaningful way) upon the state and control at the start of phase b. Denoting the number of linkage constraints on the state and control between phases a and b by $l_{(a)}^{(b)}$, and the corresponding linkage function by $\boldsymbol{L}_{(a)}^{(b)}$ where

$$\boldsymbol{L}_{(a)}^{(b)} : \mathbb{R}^{n^{(a)}} \times \mathbb{R}^{m^{(a)}} \times \mathbb{R}^{n^{(b)}} \times \mathbb{R}^{m^{(b)}} \longrightarrow \mathbb{R}^{l_{(a)}^{(b)}} \tag{33}$$

the following linkage constraints are enforced on the state and control at the end of phase a and the state and control at the start of phase b:

$$\boldsymbol{L}_{(a)}^{(b)}(\boldsymbol{x}^{(a)}(t_f^{(a)}), \boldsymbol{u}^{(a)}(t_f^{(a)}), \boldsymbol{x}^{(b)}(t_0^{(b)}), \boldsymbol{u}^{(b)}(t_0^{(b)})) = \boldsymbol{0} \tag{34}$$

It is noted that the approach developed in this subsection for linking phases was inspired by the description of linkage conditions given in the monograph by Betts,[6] a detailed understanding of the approach used in the SOCS software,[4] and numerous personal conversations with Betts.[7]

Theoretical and practical issues regarding linkage conditions

It is important to understand both the mathematical and practical nature of Equation (32) and Equation (34). Mathematically, it is important to recognize that Equations (32) and (34) preserve the Principle of Optimality.[2] Quoting from Ref. 2, the Principle of Optimality states the following:

> An optimal policy has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision

It is seen that the linkage conditions of Equation (32) and Equation (34) do not violate the Principle of Optimality because the time, state, and control that link two phases a and b are not specified *a priori*. Instead, the optimal time, optimal state, and optimal control at a phase boundary are determined by finding a solution to the optimal control problem. From a practical standpoint, Equation (32) and Equation (34) allow for flexibility in the formulation of the optimal control problem because the models for the state, control, dynamics, and path constraints in phase a can be different from the corresponding models in phase b. Consequently, the linkage constraints between phases a and b may involve a (generally nonlinear) transformation of the state and/or the control. Furthermore, Equation (34) does *not* imply continuity in either the state or the control at a phase boundary nor does Equation (34) imply that all of the components of the state and control are linked between adjacent phases. In fact, for some problems the state may necessarily be discontinuous (e.g. jettison of a mass between stages of a launch vehicle ascent) or it may not be sensible sense to link the controls (e.g. transition from atmospheric flight with aerodynamic control to exoatmospheric flight with no control).

Mathematical Description of a non-sequential Multiple-phase optimal control problem

The R-phase optimal control problem is now described using the results of the previous two subsections. Minimize the cost functional

$$J = \sum_{r=1}^{R} \mathcal{J}^{(r)} \tag{35}$$

subject to the dynamic constraints

$$\begin{aligned} \frac{d\boldsymbol{x}^{(r)}}{dt^{(r)}} &= \boldsymbol{f}^{(r)}(\boldsymbol{x}^{(r)}(t^{(r)}), \boldsymbol{u}^{(r)}(t^{(r)}), t^{(r)}) \\ &, \; r = 1, \ldots, R \, , \; t^{(r)} \in \left[t_0^{(r)}, t_f^{(r)}\right] \end{aligned} \tag{36}$$

the path constraints

$$\begin{aligned} \boldsymbol{g}_{\min}^{(r)} \leq \boldsymbol{g}^{(r)}(\boldsymbol{x}^{(r)}(t^{(r)}), \boldsymbol{u}^{(r)}(t^{(r)}), t^{(r)}) \leq \boldsymbol{g}_{\max}^{(r)}, \\ r = 1, \ldots, R \, , \; t^{(r)} \in \left[t_0^{(r)}, t_f^{(r)}\right] \end{aligned} \tag{37}$$

the initial and terminal constraints

$$\begin{aligned} \boldsymbol{h}_{0,\min}^{(r)} \leq \boldsymbol{h}_0^{(r)}(\boldsymbol{x}^{(r)}(t_0), t_0^{(r)}) \leq \boldsymbol{h}_{0,\max}^{(r)} \\ \boldsymbol{h}_{f,\min}^{(r)} \leq \boldsymbol{h}_f^{(r)}(\boldsymbol{x}^{(r)}(t_f), t_f^{(r)}) \leq \boldsymbol{h}_{f,\max}^{(r)} \\ r = 1, \ldots, R \end{aligned} \tag{38}$$

and the linkage constraints

$$\left.\begin{aligned} t_f^{(a_s)} - t_0^{(b_s)} &= 0 \\ \boldsymbol{L}_{(a_s)}^{(b_s)}(\boldsymbol{x}^{(a_s)}(t_f^{(a_s)}), \boldsymbol{u}^{(a_s)}(t_f^{(a_s)}), & \\ \boldsymbol{x}^{(b_s)}(t_0^{(b_s)}), \boldsymbol{u}^{(b_s)}(t_0^{(b_s)}) &= \boldsymbol{0} \end{aligned}\right\}, \quad s = 1, \ldots, S, \; a_s \neq b_s \tag{39}$$

where S is the number of phases that are to be linked. It is noted that, for the case where only adjacent phases are linked (i.e. a sequential multiple-phase optimal control problem), we have that

$$\begin{array}{rcl} a_s & = & s \\ b_s & = & s+1 \end{array}, \; s = 1, \ldots, R-1 \tag{40}$$

EXTENSION OF PSEUDOSPECTRAL LEGENDRE METHOD TO NON-SEQUENTIAL MULTIPLE-PHASE OPTIMAL CONTROL PROBLEMS

The non-sequential multiple-phase optimal control problem described int the previous section is now transcribed to a discrete NLP via an extension of the single-phase pseudospectral Legendre method of Ref. 10. First, let R be the number of phases and let $N^{(r)}$ be the number of LGL points (nodes) in phase $r \in [1, \ldots, R]$. Next, let $\tau_k^{(r)}$, $w_k^{(r)}$, and $\boldsymbol{D}^{(r)}$, be the LGL points, weights, and differentiation matrix, respectively, in phase $r \in [1, \ldots, R]$ corresponding the choice of $N^{(r)}$. Finally, let $\boldsymbol{x}_k^{(r)}$ and $\boldsymbol{u}_k^{(r)}$, $k = 0, 1, \ldots, N^{(r)}$ be the state and control in phase $r \in [1, \ldots, R]$ at each of the LGL points. Then, using the results of the previous two subsections and the results of Equations (17)-(22), the cost function in phase $r \in [1, \ldots, R]$ is given as

$$\begin{aligned} J^{(r)} &= \varphi^{(r)}(\boldsymbol{x}_0^{(r)}, t_0^{(r)}, \boldsymbol{x}_{N^{(r)}}^{(r)}, t_f^{(r)}) \\ &\quad + \frac{t_f^{(r)} - t_0^{(r)}}{2} \sum_{k=0}^{N^{(r)}} \mathcal{L}^{(r)}(\boldsymbol{x}_k^{(r)}, \boldsymbol{u}_k^{(r)}, \tau_k^{(r)}, t_0^{(r)}, t_f^{(r)}) w_k^{(r)}, \\ &\qquad\qquad r = 1, \ldots, R \end{aligned} \tag{41}$$

Furthermore, the discretized form of the dynamics and path constraints are given, respectively, as

$$\begin{aligned} &\sum_{l=0}^{N^{(r)}} D_{kl}^{(r)} \boldsymbol{x}_l^{(r)} - \frac{t_f^{(r)} - t_0^{(r)}}{2} \boldsymbol{f}^{(r)}(\boldsymbol{x}_k^{(r)}, \boldsymbol{u}_k^{(r)}, \tau_k^{(r)}, t_0^{(r)}, t_f^{(r)}) \\ &= \boldsymbol{0}, \\ &k = 0, 1, \ldots, N^{(r)} \\ &r = 1, \ldots, R \end{aligned} \tag{42}$$

$$\begin{aligned} &\boldsymbol{g}_{\min}^{(r)} \leq \boldsymbol{g}^{(r)}\left(\boldsymbol{x}_k^{(r)}, \boldsymbol{u}_k^{(r)}, \tau_k^{(r)}, t_0^{(r)}, t_f^{(r)}\right) \leq \boldsymbol{g}_{\max}^{(r)}, \\ &\qquad k = 0, 1, \ldots, N^{(r)}, \; r = 1, \ldots, R \end{aligned} \tag{43}$$

Furthermore, the initial and terminal constraints are given as

$$\begin{array}{rcl} \boldsymbol{h}_{0,\min}^{(r)} & \leq \boldsymbol{h}_0^{(r)}(\boldsymbol{x}_0^{(r)}, t_0^{(r)}) \leq & \boldsymbol{h}_{0,\max}^{(r)} \\ \boldsymbol{h}_{f,\min}^{(r)} & \leq \boldsymbol{h}_f^{(r)}(\boldsymbol{x}_{N^{(r)}}^{(r)}, t_f^{(r)}) \leq & \boldsymbol{h}_{f,\max}^{(r)} \end{array}, \quad r = 1, \ldots, R \tag{44}$$

Finally, the discretized form of the linkage constraints is given as

$$\begin{array}{rcl} t_f^{(a_s)} - t_0^{(b_s)} & = & 0 \\ \boldsymbol{L}(\boldsymbol{x}_{N^{(a_s)}}^{(a_s)}, \boldsymbol{u}_{N^{(a_s)}}^{(a_s)}, \boldsymbol{x}_0^{(b_s)}, \boldsymbol{u}_0^{(b_s)}) & = & \boldsymbol{0} \end{array}, \quad s = 1, \ldots, S, \; a_s \neq b_s \tag{45}$$

The NLP that arises from Equations (41)-(45) is given as follows. Minimize the cost function

$$\mathcal{J} = \sum_{r=1}^{R} J^{(r)} \tag{46}$$

where $J^{(r)}$ is from Equation (41), subject to the constraints of Equations (42)-(45). It is noted that the aforementioned NLP has

$$\sum_{r=1}^{R} \left((N^{(r)} + 1)(n^{(r)} + m^{(r)}) + 2 \right)$$

variables and

$$\sum_{r=1}^{R} \left((N^{(r)} + 1)(n^{(r)} + p^{(r)}) + q_0^{(r)} + q_1^{(r)} \right) + \sum_{r=1}^{R-1} (l_{(r)}^{(r+1)} + 1)$$

constraints.

PRACTICAL IMPLEMENTATION ISSUES

Several practical implementation issues arise in order to implement the multiple-phase pseudospectral Legendre method in a computationally efficient and robust manner. For completeness, some of these issues are now discussed.

First, because the constraint Jacobian is sparse, use of a dense NLP solver (e.g. NPSOL[11]) will in general be significantly less computationally efficient than the use of a sparse NLP solver (e.g. SPRNLP[3] or

SNOPT[12,13]). Furthermore, from the detailed description of the dependencies it is relatively straightforward to implement a reusable procedure to determine the sparsity structure of the Jacobian. This procedure requires that the user provide as an input a Kronecker delta function matrix that specifies the dependencies of the continuous-time differential equations and continuous-time path constraints on the components of the continuous-time state and continuous-time control. If a particular component of either $\boldsymbol{f}^{(r)}$ or $\boldsymbol{g}^{(r)}$ depends on a particular component of $\boldsymbol{x}^{(r)}$ or $\boldsymbol{u}^{(r)}$, then that element in the Kronecker delta function matrix is unity, otherwise it is zero. Then the corresponding off-diagonal block in phase r is either the $(N^{(r)}+1)\times(N^{(r)}+1)$ identity matrix or the $(N^{(r)}+1)\times(N^{(r)}+1)$ zero matrix.

A second issue pertains to the structure of the diagonal blocks of the NLP Jacobian that correspond to the differentiation matrix. Examining the structure of the NLP constraints that arise from the pseudospectral discretization of the differential equations, it can be seen that any element in the constraint Jacobian that corresponds to an off-diagonal element of a differentiation matrix is *constant*, i.e. the differential equation constraints can be separated into non-overlapping linear and nonlinear parts. Many NLP solvers, including SNOPT,[12,13] have the ability to exploit such non-overlapping structure in order to reduce the amount of computation required to compute the Jacobian. In such cases, the constant derivatives are stored in the same matrix where the derivatives of the linear constraints are stored. Consequently, these derivatives are never computed, but are used whenever the constraint Jacobian is required by the optimizer.

A third important, but possibly subtle, issue pertains to how to properly constrain the independent variable. In particular, constraints on the independent variables arise for three main reasons: (1) monotonicity requirements, (2) bounds on the length of each phase, and (3) the linkage constraints connecting adjacent phases. First, in most problems of practical interest, the independent variable will be monotonic. Monotonicity is ensured by placing linear inequality constraints on the start and terminal values of the independent variable within each phase, i.e. by placing linear constraints of the form

$$\begin{array}{rcll} t_f^{(r)} & \geq & t_0^{(r)} & \text{for an increasing independent variable} \\ t_f^{(r)} & \leq & t_0^{(r)} & \text{for a decreasing independent variable} \\ & r=1,\ldots,R & & \end{array} \tag{47}$$

Second, the length of each phase can be bounded by placing linear inequality constraints of the form

$$T_l^{(r)} \leq t_f^{(r)} - t_0^{(r)} \leq T_u^{(r)} \ , \ r=1,\ldots,R \tag{48}$$

where $T_l^{(r)}$ and $T_u^{(r)}$ are the lower and upper bounds, respectively, on the length of phase $r \in [1,\ldots,R]$. Third, the linkage conditions on the independent variable are implemented as linear equality constraints. In the case where it is desired to use different scales for the independent variable in different phases, the linkage conditions of Equation (32) can be easily modified as follows:

$$t_f^{(a_s)} = c_{(a_s)}^{(b_s)} t_0^{(b_s)}, \ s=1,\ldots,S \tag{49}$$

where $c_{(a_s)}^{(b_s)}$ is a constant defining the ratio of the scales of the independent variable between phase a_s and phase b_s. Finally, it is noted that, from experience, it has been found that if a linear constraint is assumed to be *nonlinear*, the amount of computational effort required by the NLP solver increases significantly. In some instances not taking advantage of linear constraints has led to the optimizer not converging to a solution.

SOFTWARE IMPLEMENTATION OF METHOD

The aforementioned multiple-phase pseudospectral Legendre method has been coded in a general purpose software program using the MATLAB[1] MEX interface version of the NLP Solver SNOPT[13] and developed by Gill.[13] The software interface is devised so that the user must specify the phase structure of the continuous-time optimal control problem and provide functions in each phase for relevant quantities (i.e. cost functional, differential equations, path constraints, interior point constraints, and linkage constraints). The program then transcribes this information internally into an NLP. The NLP has the structure as described in the previous section. In order to take maximum advantage of the sparsity of the constraint Jacobian of the NLP, the user is required to provide a Kronecker delta function matrix that specifies the dependencies of the continuous-time differential equations and continuous-time path constraints on the continuous-time state and control in each phase.

[1]MATLAB is a registered trademark of The Mathworks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098

This Kronecker delta function matrix is then used to determine those blocks in the constraint Jacobian (see previous section) that are zero and those blocks that are the identity matrix. Also, the software allows the user to either have all derivatives computed numerically or to provide analytic derivatives. If analytic derivatives are chosen, the user must provide the nonzero derivatives associated with the first-order information from the continuous-time optimal control problem along with the nonzero derivatives of the linkage constraints.

The procedure used to assign the analytic derivatives to the NLP Jacobian is as follows. First, before the NLP is solved, a subroutine is called that determines the coordinates in the NLP constraint Jacobian of the derivatives of the various functions in each phase of the NLP. At the interface level, the user must provide a set of matrices that correspond to the Jacobians of each of the functions in each phase of the problem. Taking advantage of the ability to write vectorized code in MATLAB, these matrices are coded in a manner that maintains a great deal of similarity to the mathematical from of the derivatives. Then, using the pre-determined coordinates for the various derivatives, the code is structured to assign (in a vectorized manner to maximize the efficiency of the MATLAB assignments) the derivatives obtained from the user-supplied matrices into the proper elements of the NLP Jacobian. It is noted that this implementation of analytic derivatives has proven to significantly reduce computation time.

APPLICATION OF MULTIPLE-PHASE PSEUDOSPECTRAL LEGENDRE METHOD

The multiple-phase pseudospectral Legendre method is now demonstrated on an example. In particular, the following example illustrates the ability to change the number of controls and the number of path constraints between phases. It is noted that analytic derivatives were used to solve the example problem.

Reusable launch vehicle de-orbit, coast, and entry

As an example, consider the de-orbit, coast, and reentry of a reusable launch vehicle. During de-orbit it is assumed that the vehicle is in powered exo-atmospheric flight. During coast it is assumed that the vehicle is in unpowered exo-atmospheric flight. During entry it is assumed that the vehicle is in unpowered atmospheric flight. The terminal times of the de-orbit, coast, and entry are unspecified, but the vehicle is constrained to terminate the coast at a specified radius and speed and is constrained to terminate the entry at a specified radius, speed, and flight path angle. Furthermore, the amount of mass that the vehicle can burn during de-orbit is constrained. The objective is to maximize the crossrange at the terminus of the entry phase while meeting the interior and terminal constraints.

The problem described above can be modeled as a three-phase optimal control problem as follows. During de-orbit the equations of motion are given in Earth-centered inertial (ECI) Cartesian coordinates as

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= T\boldsymbol{u}_{eci}/m - \mu\boldsymbol{r}/\|\boldsymbol{r}\|^3 \\ \dot{m} &= -T/(g_0 I_{sp}) \end{aligned} \tag{50}$$

where $\boldsymbol{r} = (x, y, z)$ is the ECI position, $\boldsymbol{v} = (v_x, v_y, v_z)$ is the ECI velocity, m is the vehicle mass, T is the magnitude of the thrust force, $\boldsymbol{u}_{eci}$ is the ECI thrust direction, μ is the Earth gravitational parameter, g_0 is the magnitude of the gravitational acceleration at sea level, and I_{sp} is the specific impulse of the engine during de-orbit. It is assumed during de-orbit that T is constant. Furthermore, the ECI thrust direction, $\boldsymbol{u}_{eci}$ is computed as follows. First, the vector $\boldsymbol{u} = (u_1, u_2, u_3)$ is defined in a velocity coordinate system (see Appendix). The vector $\boldsymbol{u}$ is then transformed from the velocity coordinate system to the ECI coordinate system via the transformation $\boldsymbol{E}_{v2e}$ (see Appendix) as follows:

$$\boldsymbol{u}_{eci} = \boldsymbol{E}_{v2e}\boldsymbol{u} \tag{51}$$

Finally, because $\boldsymbol{u}$ must be a unit vector, the following equality path constraint is imposed on $\boldsymbol{u}$ during de-orbit:

$$\boldsymbol{u} \cdot \boldsymbol{u} = u_1^2 + u_2^2 + u_3^2 = 1 \tag{52}$$

The equations of motion during the coast are given in ECI coordinates as

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= -\mu\boldsymbol{r}/\|\boldsymbol{r}\|^3 \\ \dot{m} &= 0 \end{aligned} \tag{53}$$

Finally, the equations of motion during entry are mod-

eled in spherical coordinates as[22]

$$\begin{aligned}
\dot{r} &= v\sin\gamma \\
\dot{\theta} &= v\cos\gamma\cos\psi/(r\cos\phi) \\
\dot{\phi} &= v\cos\gamma\sin\psi/r \\
\dot{v} &= -qSC_D/m - g\sin\gamma \\
\dot{\gamma} &= qSC_L w_1/(mv) - (g/v - v/r)\cos\gamma \\
\dot{\psi} &= qSC_L w_2/(mv\cos\gamma) - (v/r)\cos\gamma\cos\psi\tan\phi \\
\dot{m} &= 0
\end{aligned} \tag{54}$$

where r is the geocentric radius, θ is the longitude, ϕ is the geocentric latitude, v is the speed, γ is the flight path angle, ψ is the heading angle, $g = \mu/r^2$ is the magnitude of the gravitational acceleration, $q = \rho v^2/2$ is the dynamic pressure, ρ is the atmospheric density, C_D is the coefficient of drag, C_L is the coefficient of lift, and S is the vehicle reference area. The density and aerodynamic models used in this example are taken from.[25] The controls in the entry phase are the angle of attack, α, and the two components of the lift direction, $\boldsymbol{w} = (w_1, w_2)$. Similar to the de-orbit phase, the vector $\boldsymbol{w}$ must be a unit vector. Consequently, the following equality path constraint is imposed on $\boldsymbol{w}$ during entry:

$$\boldsymbol{w}\cdot\boldsymbol{w} = w_1^2 + w_2^2 = 1 \tag{55}$$

The initial and terminal constraints, interior point constraints, linkage constraints, and bounds are defined as follows. The initial conditions (at time $t = t_0^{(1)} = 0$) correspond to an equatorial circular orbit at an altitude $h(t_0^{(1)}) = h_0 = 378000$ m and are given as

$$\begin{aligned}
\boldsymbol{r}(t_0^{(1)}) &= (r_0, 0, 0) = (x_0, y_0, z_0) \equiv \boldsymbol{r}_0 \\
\boldsymbol{v}(t_0^{(1)}) &= (0, v_0, 0) = (v_{x0}, v_{y0}, v_{z0}) \equiv \boldsymbol{v}_0 \\
m(t_0^{(1)}) &= 97419.6 \text{ kg} \equiv m_0
\end{aligned} \tag{56}$$

where $r_0 = h_0 + R_e$, $v_0 = \sqrt{\mu/r_0}$, and $R_e = 6378145$ m is the radius of the Earth. Furthermore, the following inequality constraint is placed on the amount of mass that can be burned during the de-orbit phase:

$$m \geq m_{1,\min} = 77482.5 \text{ kg} \tag{57}$$

where $m_{1,\min}$ is the maximum amount of mass that can be burned during the de-orbit. Denoting $t_f^{(1)}$ as the (unspecified) terminal time of the de-orbit and $t_0^{(2)}$ as the (unspecified) start time of the coast, the de-orbit and coast phases are linked via the following linkage conditions:

$$\begin{aligned}
t_f^{(1)} - t_0^{(2)} &= 0 \\
\boldsymbol{r}(t_f^{(1)}) - \boldsymbol{r}(t_0^{(2)}) &= \boldsymbol{0} \\
\boldsymbol{v}(t_f^{(1)}) - \boldsymbol{v}(t_0^{(2)}) &= \boldsymbol{0} \\
m(t_f^{(1)}) - m(t_0^{(2)}) &= 0
\end{aligned} \tag{58}$$

Denoting $t_f^{(2)} > t_0^{(2)}$ as the (unspecified) terminal time of the coast phase, the coast phase terminates when the following two conditions are met:

$$\begin{aligned}
r(t_f^{(2)}) = \|\boldsymbol{r}(t_f^{(2)})\|_2 &= 121920 \text{ m} + R_e \equiv r_{2f} \\
v(t_f^{(2)}) = \|\boldsymbol{v}(t_f^{(2)})\|_2 &= 7802 \text{ m/s} \equiv v_{2f}
\end{aligned} \tag{59}$$

The coast phase and entry phase are linked as follows. The state at the end of the coast phase, denoted $\boldsymbol{x}(t_f^{(2)})$, is given as

$$\boldsymbol{x}(t_f^{(2)}) = \left(\boldsymbol{r}(t_f^{(2)}), \boldsymbol{v}(t_f^{(2)}), m(t_f^{(2)})\right) \tag{60}$$

Denoting $t_0^{(3)}$ as the start time for the entry phase, the state in spherical coordinates at the start of entry, denoted $\boldsymbol{y}(t_0^{(3)})$, is given as

$$\begin{aligned}
\boldsymbol{y}(t_0^{(3)}) = (&r(t_0^{(3)}), \theta(t_0^{(3)}), \phi(t_0^{(3)}), \\
&v(t_0^{(3)}), \gamma(t_0^{(3)}), \psi(t_0^{(3)}), m(t_0^{(3)}))
\end{aligned} \tag{61}$$

The Cartesian quantities $\boldsymbol{r}(t_f^{(2)})$ and $\boldsymbol{v}(t_f^{(2)})$ are transformed to spherical coordinates via the transformation T_{c2s} (see Appendix) as

$$\begin{aligned}
&\left(r(t_f^{(2)}), \theta(t_f^{(2)}), \phi(t_f^{(2)}), v(t_f^{(2)}), \gamma(t_f^{(2)}), \psi(t_f^{(2)})\right) \\
&= T_{c2s}(\boldsymbol{r}(t_f^{(2)}), \boldsymbol{v}(t_f^{(2)}))
\end{aligned} \tag{62}$$

The transformed state at the end of the coast, denoted $\tilde{\boldsymbol{y}}(t_f^{(2)})$, is then given as

$$\begin{aligned}
\tilde{\boldsymbol{y}}(t_f^{(2)}) = (&r(t_f^{(2)}), \theta(t_f^{(2)}), \phi(t_f^{(2)}), \\
&v(t_f^{(2)}), \gamma(t_f^{(2)}), \psi(t_f^{(2)}), m(t_f^{(2)}))
\end{aligned} \tag{63}$$

The linkage conditions between the coast phase and the entry phase are then given as

$$\begin{aligned}
t_f^{(2)} - t_0^{(3)} &= 0 \\
\tilde{\boldsymbol{y}}(t_f^{(2)}) - \boldsymbol{y}(t_0^{(3)}) &= \boldsymbol{0}
\end{aligned} \tag{64}$$

Finally, denoting the (unspecified) terminal time of the entry phase as $t_f^{(3)} > t_0^{(3)}$, the entry phase terminates when the following three conditions are met:[25]

$$\begin{aligned}
r(t_f^{(3)}) &= 24384 \text{ m} + R_e = r_f \\
v(t_f^{(3)}) &= 762 \text{ m/s} = v_f \\
\gamma(t_f^{(3)}) &= -5 \text{ deg} = \gamma_f
\end{aligned} \tag{65}$$

Note that, because the controls are different in each phase, they are not linked. Furthermore, the following

bounds are placed on the variables in each phase as follows. In the de-orbit phase the bounds are

$$\begin{array}{rcc}
-\begin{bmatrix} r_0 \\ r_0 \\ r_0 \end{bmatrix} \leq & (x,y,z) \leq & \begin{bmatrix} r_0 \\ r_0 \\ -r_0 \end{bmatrix} \\
-\begin{bmatrix} 10v_0 \\ 10v_0 \\ 10v_0 \end{bmatrix} \leq & (v_x,v_y,v_z) \leq & \begin{bmatrix} 10v_0 \\ 10v_0 \\ 10v_0 \end{bmatrix} \\
77483 \text{ kg} \leq & m & \leq 97420 \text{ kg} \\
-1 \leq & u_1 & \leq 1 \\
-1 \leq & u_2 & \leq 1 \\
-1 \leq & u_3 & \leq 1 \\
T & = & 26689 \text{ N}
\end{array} \tag{66}$$

In the coast phase the bounds are

$$\begin{array}{rcc}
-\begin{bmatrix} r_0 \\ r_0 \\ r_0 \end{bmatrix} - \leq & (x,y,z) & \leq \begin{bmatrix} r_0 \\ r_0 \\ -r_0 \end{bmatrix} \\
-\begin{bmatrix} 10v_0 \\ 10v_0 \\ 10v_0 \end{bmatrix} \leq & (v_x,v_y,v_z) \leq & \begin{bmatrix} 10v_0 \\ 10v_0 \\ 10v_0 \end{bmatrix} \\
77483 \text{ kg} \leq & m & \leq 97420 \text{ kg}
\end{array} \tag{67}$$

In the entry phase the bounds are

$$\begin{array}{rcccc}
R_e & \leq & r & \leq & 3r_1 \\
-720 \text{ deg} & \leq & \theta & \leq & 720 \text{ deg} \\
-80 \text{ deg} & \leq & \phi & \leq & 80 \text{ deg} \\
100 \text{ m/s} & \leq & v & \leq & 10v_2 \\
-70 \text{ deg} & \leq & \gamma & \leq & 70 \text{ deg} \\
-360 \text{ deg} & \leq & \psi & \leq & 360 \text{ deg} \\
77483 \text{ kg} & \leq & m & \leq & 77483 \text{ kg} \\
0 & \leq & \alpha & \leq & 40 \text{ deg} \\
-1 & \leq & w_1 & \leq & 1 \\
-1 & \leq & w_2 & \leq & 1
\end{array} \tag{68}$$

Because the initial condition for this problem is at the equator, the objective of maximizing the crossrange at the end of entry is equivalent to maximizing the latitude at the end of entry, i.e. the objective is to maximize

$$J = \phi(t_f^{(3)}) \tag{69}$$

Furthermore, using R_e as the reference length, the downrange and crossrange distances are computed as

$$\begin{array}{rcl} d & = & R_e\theta \\ c & = & R_e\phi \end{array} \tag{70}$$

where θ and ϕ are given in radians. It is noted that for the de-orbit and coast phases, the angles θ and ϕ are computed from $\boldsymbol{r}$ and $\boldsymbol{v}$ using the transformation T_{c2s} given in the Appendix. Lastly, the three-phase optimal control problem was solved in a scaled set of units such that the length scale was R_e, the speed scale was $\sqrt{\mu/R_e}$, and the mass scale was m_0. All other scale factors were derived so as to maintain a canonical transformation.

The maximum crossrange problem described above was solved as follows. First, 20 nodes were chosen for the de-orbit and coast phases while 40 nodes was chosen for the entry phase. In addition, default tolerances were used for SNOPT and the following initial guess was provided. In the de-orbit phase the initial guess was

$$\begin{array}{c}
t_0^{(1)} = 0 \text{ s} \quad , \quad t_f^{(1)} = 900 \text{ s} \\
\boldsymbol{r}(t) = \boldsymbol{r}_0 \quad , \quad \boldsymbol{v}(t) = \boldsymbol{v}_0 \\
m(t) = m_0 \\
u_1(t) = 1 \quad , \quad u_2(t) = 0 \quad , \quad u_3(t) = 0
\end{array} \tag{71}$$

In the coast phase the initial guess was

$$\begin{array}{ccc}
t_0^{(2)} = t_f^{(1)} = 900 \text{ s} & , & t_f^{(2)} = 910 \text{ s} \\
\boldsymbol{r}(t) = \boldsymbol{r}_0 & , & \boldsymbol{v}(t) = \boldsymbol{v}_0 \\
m(t) = m_0 & &
\end{array} \tag{72}$$

In the entry phase the initial guess was

$$\begin{array}{lcl}
t_0^{(3)} = t_f^{(2)} = 910 \text{ s} & , & t_f^{(3)} = 1800 \text{ s} \\
r(t) = r_f & , & \theta(t) = 0 \\
\phi(t) = 0 & , & v(t) = v_f \\
\gamma(t) = 0 & , & \psi(t) = 0 \\
\alpha(t) = 0 & , & \sigma(t) = 0 \\
w_1(t) = 1 & , & w_2(t) = 0
\end{array} \tag{73}$$

Using the above initialization, the optimized objective function value was found to be $\phi^*(t_f^{(3)}) = 37.943$ deg (approximately 4223 km). The key results are summarized in Figs. 4-5 where the Euler pitch and yaw angles, denoted by β and χ, respectively, are defined in the velocity frame (see Appendix) and are computed from the unit thrust direction $\boldsymbol{u}$ as

$$\begin{array}{rcl} \beta & = & \sin^{-1}(u_3) \\ \chi & = & \tan^{-1}(u_2, u_1) \end{array} \tag{74}$$

Similarly, the bank angle, σ, shown in Fig. 5 is computed using the components of the lift direction, $\boldsymbol{w}$, as

$$\sigma = \tan^{-1}(w_2, w_1) \tag{75}$$

In Equation (74) and Equation (75) it is noted that the functions $\sin^{-1}(\cdot)$ and $\tan^{-1}(\cdot,\cdot)$ are the inverse sine and four-quadrant inverse tangent, respectively. Finally, the feasibility of the obtained solution was assessed by numerically integrating the trajectory from the initial condition using the optimized controls.

COMMENTS ON RESULTS

The previous example demonstrates three key features of the multiple-phase pseudospectral Legendre method. First, the example demonstrates the ability to use different coordinate systems in different phases. Furthermore, the example demonstrates the ability to change path constraints between phases and shows the generality of the linkage constraints.

CONCLUSIONS

A pseudospectral Legendre method for discretizing continuous-time optimal control problems has been extended to the case of non-sequential multiple-phase optimal control problems. In the extension developed here, the dimension of the state, control, and constraints can be different in each phase, thus allowing for a general formulation of the optimal control problem. The phases are linked via linkage conditions on the state, control, and independent variable. The structure of the resulting nonlinear programming (NLP) problem has been described in detail and several key implementation issues have been discussed. A particular software implementation of the method that takes advantage of the particular sparsity structure of the NLP has also been described. Finally, the method has been demonstrated on a representative example that illustrates its key features. The approach developed here offers a viable method for solving general non-sequential multiple-phase optimal control problems.

Velocity coordinates to Cartesian coordinates

Let $\boldsymbol{r}$ and $\boldsymbol{v}$ be the position and velocity, respectively, of a vehicle in Cartesian coordinates. Then the velocity coordinate system is defined by the unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ as follows:

$$\begin{array}{rcl} \boldsymbol{e}_1 &=& \boldsymbol{v}/\|\boldsymbol{v}\|_2 \\ \boldsymbol{e}_2 &=& (\boldsymbol{r}\times\boldsymbol{v})/\|\boldsymbol{r}\times\boldsymbol{v}\|_2 \\ \boldsymbol{e}_3 &=& \boldsymbol{e}_1\times\boldsymbol{e}_2 \end{array} \tag{76}$$

The transformation from velocity coordinates to Cartesian coordinates is then given as

$$\boldsymbol{E}_{v2e} = \left[\begin{array}{ccc} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 \end{array}\right] \tag{77}$$

Cartesian coordinates to spherical coordinates

Let $\boldsymbol{r} = (x, y, z)$ and $\boldsymbol{v} = (v_x, v_y, v_z)$ denote the position and velocity, respectively, of a vehicle in Cartesian coordinates. Let (r, θ, ϕ) denote the position in spherical coordinates where r is the radius, θ is the longitude, and ϕ is the latitude. Similarly, let (v, γ, ψ) denote the velocity in spherical coordinates where v is the speed, γ is the flight path angle, and ψ is the heading angle. Then the transformation from Cartesian coordinates to spherical coordinates is given as follows. The position is transformed as

$$\begin{array}{rcl} r &=& \|\boldsymbol{r}\|_2 = \sqrt{x^2+y^2+z^2} \\ \theta &=& \tan^{-1}(y, x) \\ \phi &=& \sin^{-1}(z/r) \end{array} \tag{78}$$

The velocity is transformed as follows. Let

$$\boldsymbol{T}_{s2e} = \left[\begin{array}{ccc} \boldsymbol{e}_r & \boldsymbol{e}_\theta & \boldsymbol{e}_\phi \end{array}\right] \tag{79}$$

be defined so that

$$\begin{array}{rcl} \boldsymbol{e}_r &=& \boldsymbol{r}/r \\ \boldsymbol{e}_\theta &=& (\boldsymbol{u}_z\times\boldsymbol{r})/\|\boldsymbol{u}_z\times\boldsymbol{r}\|_2 \\ \boldsymbol{e}_\phi &=& \boldsymbol{e}_r\times\boldsymbol{e}_\theta \end{array} \tag{80}$$

where $\boldsymbol{u}_z$ is the unit vector in the z direction. Then, denoting the velocity in the coordinate system $\{\boldsymbol{e}_r, \boldsymbol{e}_\theta, \boldsymbol{e}_\phi\}$ as $\boldsymbol{v}_s$, we obtain

$$\boldsymbol{v}_s = T_{s2e}^T\boldsymbol{v} = (v_{sr}, v_{s\theta}, v_{s\phi}) \tag{81}$$

We then obtain

$$\begin{array}{rcl} v &=& \|\boldsymbol{v}_s\|_2 = \sqrt{v_{sr}^2+v_{s\theta}^2+v_{s\phi}^2} \\ \gamma &=& \sin^{-1}(v_{sr}/v) \\ \psi &=& \tan^{-1}(v_{s\theta}, v_{s\phi}) \end{array} \tag{82}$$

Thus, the transformation from Cartesian coordinates to spherical coordinates, denoted T_{c2s} is given by Equations (78) and (82). Note that in the calculations above, a four-quadrant inverse tangent function is used to compute θ and ψ, resulting in θ and ψ each lying in the interval $[-\pi, \pi]$. Then θ and ψ can be modified to lie on the interval $[0, 2\pi]$ by adding 2π if either angle is less than zero that results from the four-quadrant inverse tangent function.

ACKNOWLEDGEMENTS

This research was funded by The Charles Stark Draper Laboratory, Inc. under internal research and development funds. The author acknowledges Mr. Douglas M. Fuhry for his help with the example problems presented in this paper. The author also acknowledges Dr. Owen Deutsch for providing the ascent launch vehicle models used in this research. The author also acknowledges Professor Wai-Sun Don for providing a FORTRAN 90 module for the pseudospectral computations used in this research and thanks Mr. H. Lee Norris, III, for writing an elegant MATLAB version of Professor Don's module. The author also acknowledges Dr. John T. Betts of The Boeing Company for providing the author with valuable advice regarding proper implementation in the numerical solution of multiple-phase optimal control problems. Finally, the author gratefully acknowledges Mr. Timothy Brand for his support, helpful suggestions, and guidance throughout the course of this research.

REFERENCES

[1]Athans, M. and Falb, P. L., *Optimal Control*, McGraw-Hill: New York, 1966, pp. 284-351.

[2]Bellman, R. and Kalaba, R., *Dynamic Programming and Modern Control Theory*, Academic Press: San Diego, 1966, p. 35.

[3]Betts, J. T. and Frank, P. D., "A Sparse Nonlinear Optimization Algorithm," *Journal of Optimization Theory and Applications*, Vol. 82, 1994, pp.519-541.

[4]Betts, J. T. and Huffman, W. P., "Sparse Optimal Control Software - SOCS," Mathematics and Engineering Analysis Library Report, MEA-LR-085, Boeing Information and Support Services, P. O. Box 3797, Seattle, WA, 98124-2297, 15 July 1997.

[5]Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, March-April 1998, pp. 193-207.

[6]Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics Press: Philadelphia, PA, 2001, p. 76, pp. 87-92.

[7]Betts, J. T., *Personal Communications*, 2000-2002.

[8]Bertsekas, D. P., *Nonlinear Programming*, Athena Scientific Publishing: Belmont, MA, 1999, pp. 431-445.

[9]Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A., *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.

[10]Elnagar, J., Kazemi, M. A., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793-1796.

[11]Gill, P. E., Murray W., Saunders, M. A., and Wright, M. H., "User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming", Technical Report SOL 86-2, Department of Mathematics, University of California, San Diego, 30 July 1998.

[12]Gill, P. E., Murray W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979-1006.

[13]Gill, P. E., Murray W., and Saunders, "User's Guide for SNOPT Version 6: A Fortran Package for Large-Scale Nonlinear Programming", University of California, San Diego, December 2002.

[14]Gottlieb, D., Hussaini, M. Y., and Orszag, S. A., "Theory and Applications of Spectral Methods," *Spectral Methods for PDEs*, Society for Industrial and Applied Mathematics Press, Philadelphia, PA, 1984.

[15]Rao, A. V. and Mease, K. D., "Dichotomic Basis Approach to Solving Hyper-Sensitive Optimal Control Problems", *Automatica*, Vol. 35, No. 4, April 1999, pp. 633-642.

[16]Rao, A. V. and Mease, K. D., "Eigenvector Approximate Dichotomic Basis Method for Solving Hyper-Sensitive Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 21, No. 1, 2000, pp. 1-18.

[17]Rao, A. V., Tang, S., and Hallman, W. P., "Numerical Optimization Study of Multiple-Pass Aeroassisted Orbital Transfer", *Optimal Control Applications and Methods*, Vol. 23, No. 4, July-August 2002, pp. 215-238.

[18]Fahroo, F. and Ross, I. M., "A Spectral Patching Method for Direct Trajectory Optimization," *Journal of the Astronautical Sciences*, Vol. 48, Nos. 2 and 3, April-September 2000, pp. 269-286.

[19]Fahroo, F. and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method", *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, March-April 2001, pp. 270-277.

[20]Ross, I. M. and Fahroo, F., "A Direct Method for Solving Nonsmooth Optimal Control Problems", *Proceedings of the 2002 IFAC World Conference*, Barcelona, Spain, July 2002.

[21]Ross, I. M. and Fahroo, F., "User's Manual for DIDO 2002: A MATLAB Application for Dynamic Optimization", *Naval Postgraduate School Technical Report*, NPS-AA-02-2002, June 2002.

[22]Vinh, N. X., *Optimal Trajectories in Atmospheric Flight*, Elsevier Publications: New York, 1981, pp. 47-62.

[23]Vlases, W. G., Paris, S. W., Lajoie, R. M., Martens, M. J., and Hargraves, C. R., "Optimal Trajectories by Implicit Simulation," Boeing Aerospace and Electronics, Technical Report WRDC-TR-90-3056, Wright-Patterson Air Force Base, 1990.

[24]Well, K., *Graphical Environment for Simulation and Optimization*, Department of Optimization, Guidance, and Control, Stuttgart, Germany, 2002.

[25]Zondervan, K., Bauer, T., Betts, J., and Huffman, W., "Solving the Optimal Control Problem Using a Nonlinear Programming Technique Part 3: Optimal Shuttle Reentry Trajectories", *Proceedings of the AIAA/AAS Astrodynamics Conference*, AIAA-84-2039, Seattle, WA, August 20-22, 1984.

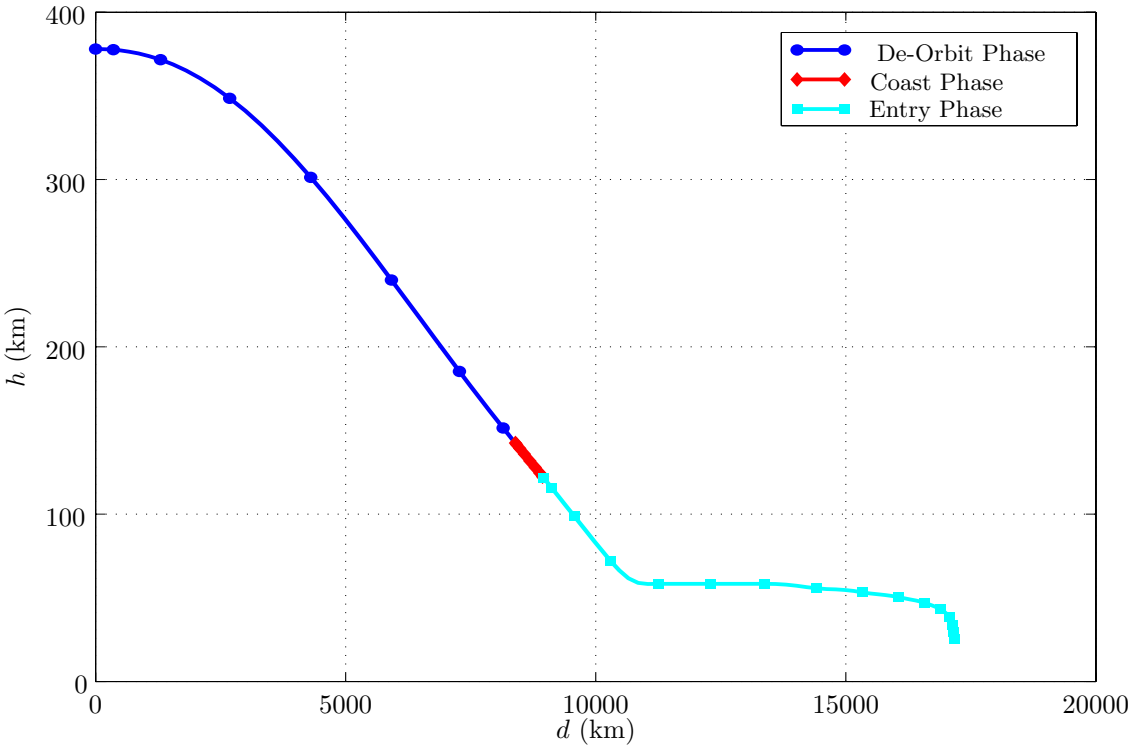

Figure 1: Altitude, h, vs. downrange, d, for reusable launch vehicle de-orbit, coast, and entry.

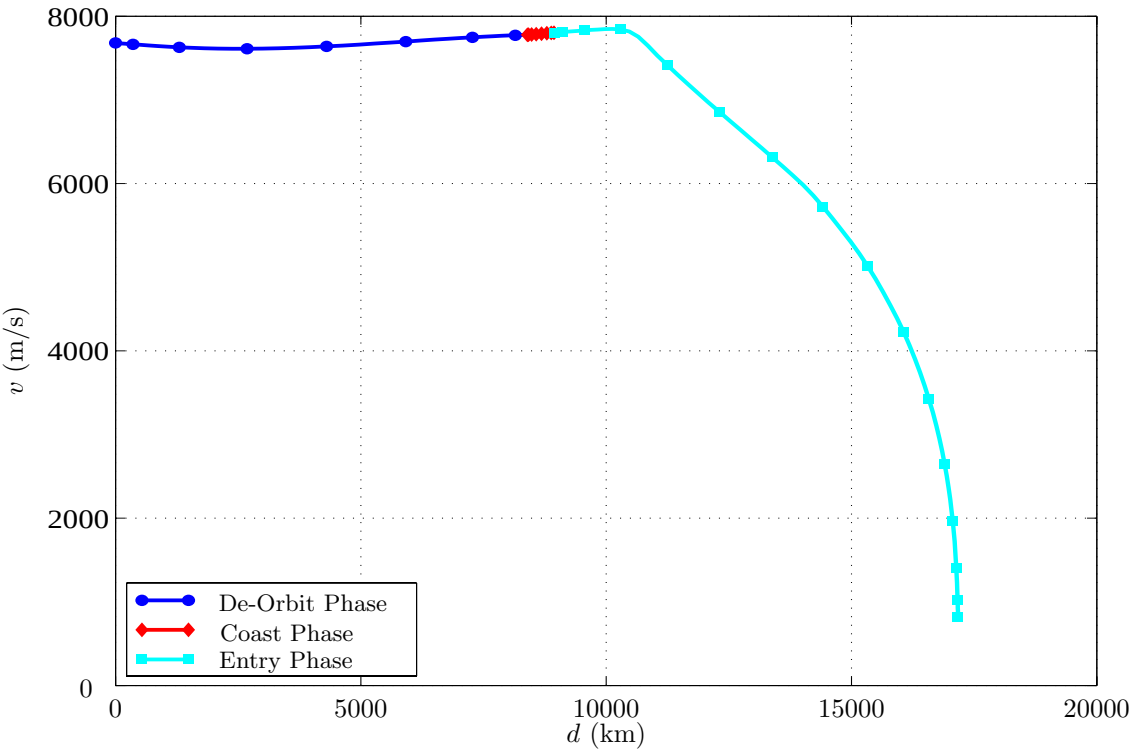

Figure 2: Speed, v, vs. downrange, d, for reusable launch vehicle de-orbit, coast, and entry.

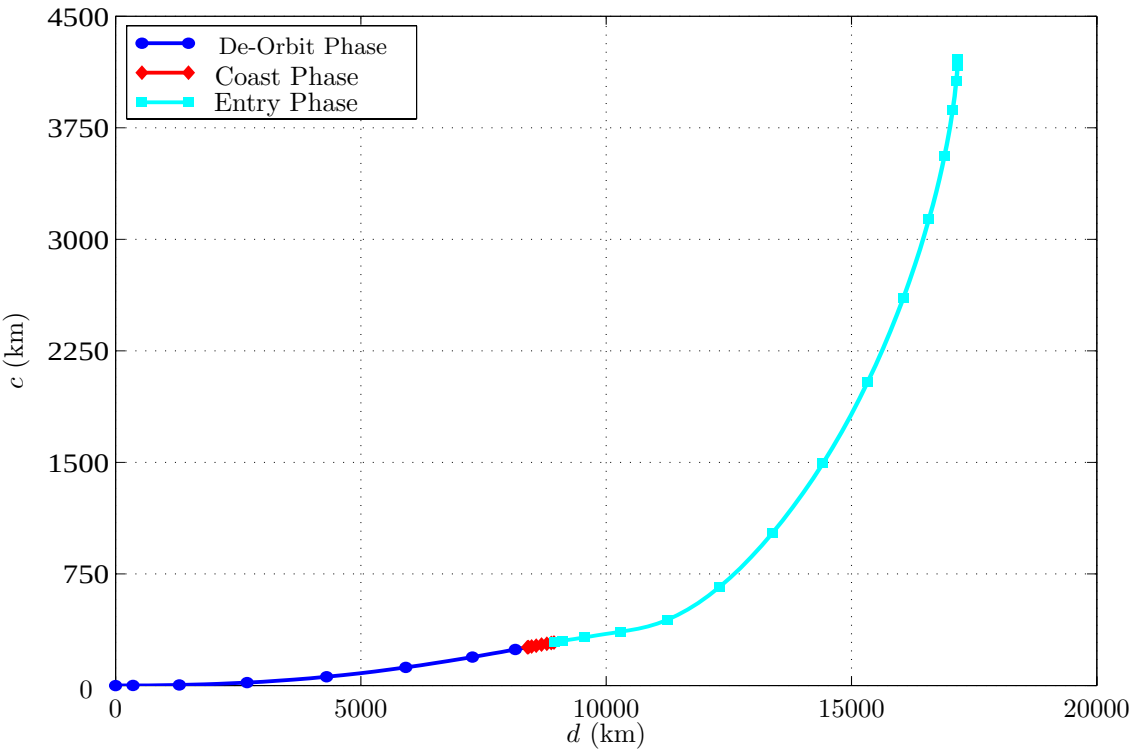

Figure 3: Crossrange, c, vs. downrange, d, for reusable launch vehicle de-orbit, coast, and entry.

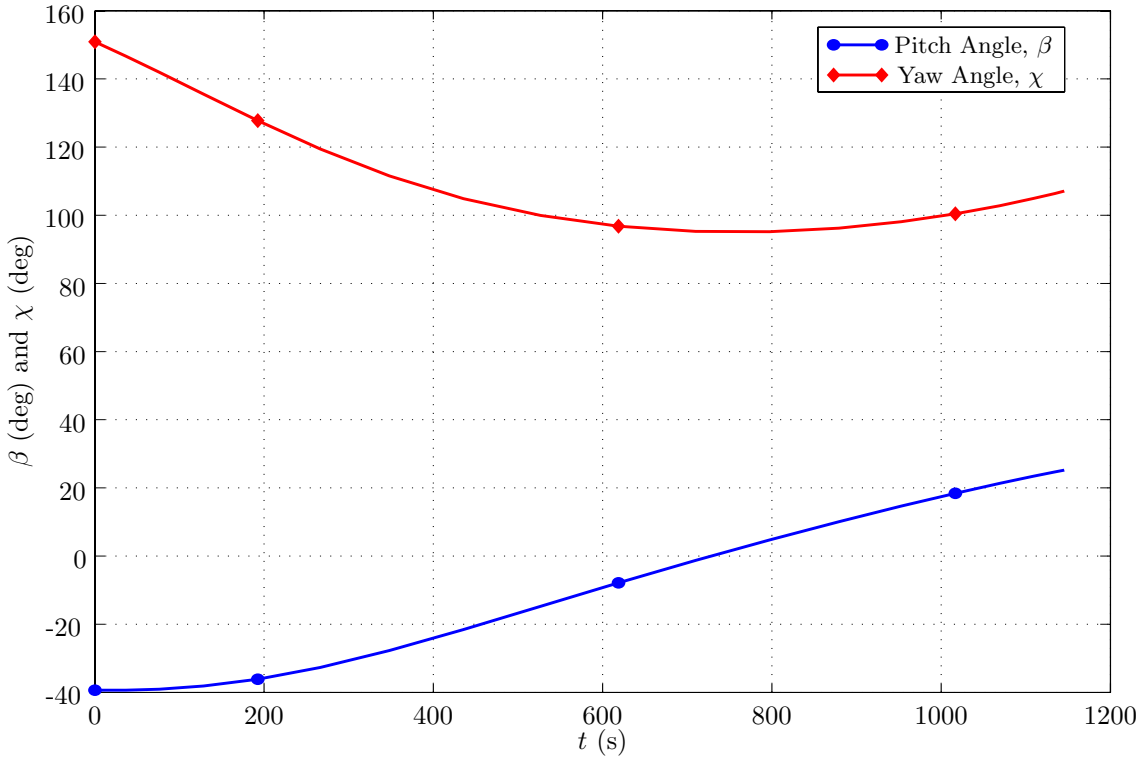

Figure 4: Pitch angle, β, and yaw angle, χ, vs. time, t, during de-orbit for reusable launch vehicle de-orbit, coast, and entry.

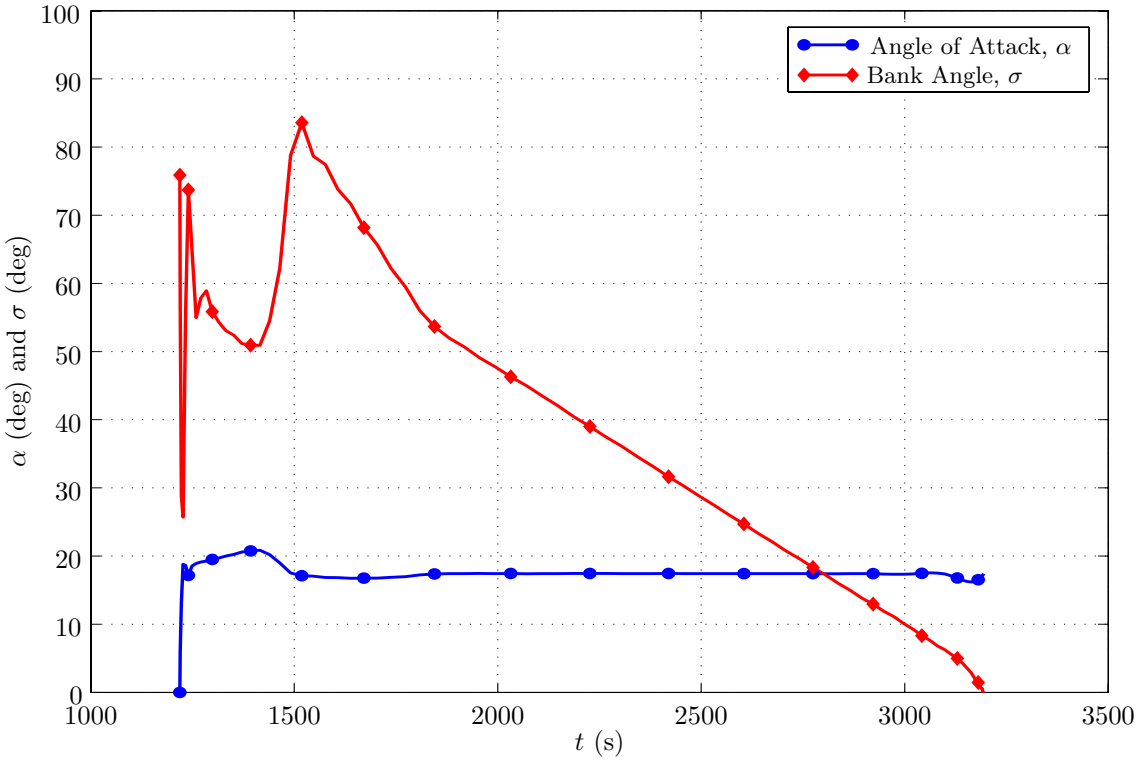

Figure 5: Angle of attack, α, and bank angle, σ, vs. time, t, during entry for reusable launch vehicle de-orbit, coast, and entry.