AIAA SciTech Forum
January 3-7, 2022, San Diego, CA & Virtual
AIAA SCITECH 2022 Forum

10.2514/6.2022-2390

Mars Entry Optimal Trajectory Generation, Guidance, and Control

Emily Palmer* and Anil V. Rao†
Department of Mechanical and Aerospace Engineering
University of Florida
Gainesville, FL 32611-6250

The problem of Mars entry trajectory optimization and guidance is considered. The vehicle is modeled as a point mass in motion over a spherical non-rotating planet. The objective is to maximize the altitude at the end of atmospheric entry subject with a terminal constraint on speed. In addition, path constraints are enforced on the sensed acceleration, heating rate, and dynamic pressure. The key features of the entry trajectory are then analyzed. Then, using the same computational approach as was used to compute the reference optimal trajectory, a method is developed for entry guidance. In particular, guidance updates are performed at constant intervals. Over each guidance cycle the dynamics are simulated using a perturbed (off-nominal) model. Then, the optimal trajectory and control are re-computed on the remaining horizon at the end of each guidance cycle. The performance of the method for guidance is then analyzed for different off-nominal coefficients of drag and guidance cycle durations.

I. Nomenclature

A	=	aerodynamic acceleration, m/s^2
C_D	=	drag coefficient of spacecraft
C_L	=	lift coefficient of spacecraft
D	=	aerodynamic drag, N
g_0	=	gravitational acceleration at the center of Earth, m/s^2
H	=	scale height, m
h	=	altitude, m
L	=	aerodynamic lift, N
$\dot{Q}$	=	heat rate, W/m^2
q	=	aerodynamic pressure, Pa
R	=	radius of Mars, m
r	=	radial distance from the center of Mars, m
r_n	=	nose radius, m
S	=	reference area of spacecraft, m^2
t	=	time, s
t_0	=	initial time, s
t_f	=	terminal time,s
v	=	spacecraft speed, m/s
x	=	state vector
γ	=	flight path angle, rad or deg
θ	=	geocentric longitude, rad or deg
σ	=	bank angle, rad or deg
ϕ	=	geocentric latitude,rad or deg
ψ	=	heading angle, rad or deg
ρ	=	Martian atmospheric density, kg/m^3
ρ_0	=	base Martian atmospheric density, kg/m^3

*Ph.D. Student, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida 32611-6250. Email: emilypalmer@ufl.edu.

†Professor, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250. E-mail: anilvrao@ufl.edu. Corresponding Author

II. Introduction

Mars entry, descent, and landing (EDL) is one of the challenging problems in Mars missions. A key aspect that makes Mars EDL such a challenging problem is the presence of a large number of constraints during entry. In particular, it is necessary to impose constraints on heating rate, sensed acceleration load, and dynamic pressure during entry. Placing a limits on these quantities ensures that the vehicle does not overheat and is not subject to excessively large structural loads during entry. In addition to designing reference trajectories that remain within acceptable constraint limits, it is also necessary to maintain these constraints when controlling the vehicle as it enters the atmosphere (that is, real-time control).

In addition to the challenge of determining entry, descent and landing reference trajectories for Mars missions, another critical aspect of a Mars mission is guidance. In particular, because of modeling errors and disturbances, the entry vehicle will not stay on the reference solution during entry unless mid-course corrections are made. Such corrections require that the control be re-computed at various points along the entry trajectory in order to steer the vehicle back on course so that it attains the terminal (target) conditions. In addition, because most reference trajectories are planned in a manner that optimizes a specified performance index, it is desirable that any mid-course corrections also be made in a manner such that the vehicle is as close to the optimal (reference) trajectory as possible.

Previous methods for guidance include proportional navigation (PN) [1, 2], linear-quadratic (LQ) methods [3–5], neighboring optimal control (NOC) [6], and other trajectory tracking methods. While these methods have been used with success in various applications, one limitation with these techniques is that neither the full nonlinear dynamics nor the full effect of the optimization criteria are taken into account in the guidance law. More recently, finite-horizon guidance and control laws have been developed where the computed control takes into account higher-order effects in the system dynamics. Finite-horizon guidance and control techniques include model predictive control[7–10] and techniques that employ nonlinear programming.

This research focuses on the application of a computational framework for the real-time solution of optimal control problems for Mars entry reference trajectory generation and guidance. The problem of interest is one where the vehicle enters the Martian atmosphere at a high altitude and high speed. In addition, during entry the vehicle is subject to constraints on sensed acceleration, heating rate, and dynamic pressure. The objective of the entry is to maximize the terminal altitude (from which a parachute could be deployed) subject to a constraint on the terminal speed. The first part of this paper focuses on determining reference optimal trajectories for this aforementioned altitude maximization problem. Specifically, the reference optimal control problem is solved using an adaptive Gaussian quadrature collocation method [11–19]. The second part of this study develops a method for optimal guidance and control using the same computational approach employed for determining the reference optimal trajectory and control. Specifically, the optimal control problem is re-solved at constant time intervals (guidance cycles), where over each guidance cycle the dynamics are simulated using an off-nominal (perturbed) dynamic model. Each time the optimal control problem is re-solved, the horizon is shorter than it was previously and the mesh in the adaptive Gaussian quadrature collocation method such that the mesh becomes progressively smaller as this horizon shrinks. The method for mesh generation and guidance used in this paper is taken from Ref. [20]. Finally, the Mars entry dynamics in this study are a three-dimensional version of the model developed in Ref [21].

This paper is organized as follows. Section III provides a description of the Bolza optimal control problem. Section IV provides a description of the Legendre-Gauss-Radau collocation method used to solve the optimal control problem numerically. Section V provides the formulation of the Mars entry optimal control problem. Section VI provides a description of the approach used to generate the reference trajectory and perform guidance updates. Section VII provides the key results obtained in this research along with a discussion of the results. Finally, Section VIII provides conclusions on this work.

III. Bolza Optimal Control Problem

Without loss of generality, consider the following general optimal control problem in Bolza form. Determine the state $\mathbf{y}(\tau) \in \mathbb{R}^{n_y}$ and the control $\mathbf{u}(\tau) \in \mathbb{R}^{n_u}$ on the domain $\tau \in [-1, +1]$, the initial time, t_0, and the terminal time t_f that minimize the cost functional

$$\mathcal{J} = \mathcal{M}(\mathbf{y}(-1), t_0, \mathbf{y}(+1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{+1} \mathcal{L}(\mathbf{y}(\tau), \mathbf{u}(\tau), t(\tau, t_0, t_f))\, d\tau, \tag{1}$$

subject to the dynamic constraints

$$\frac{d\mathbf{y}}{d\tau} - \frac{t_f - t_0}{2}\mathbf{a}(\mathbf{y}(\tau), \mathbf{u}(\tau), t(\tau, t_0, t_f)) = \mathbf{0}, \tag{2}$$

the inequality path constraints

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{y}(\tau), \mathbf{u}(\tau), t(\tau, t_0, t_f)) \leq \mathbf{c}_{\max}, \tag{3}$$

and the boundary conditions

$$\mathbf{b}_{\min} \leq \mathbf{b}(\mathbf{y}(-1), t_0, \mathbf{y}(+1), t_f) \leq \mathbf{b}_{\max}. \tag{4}$$

It is noted that the time interval $\tau \in [-1, +1]$ can be transformed to the time interval $t \in [t_0, t_f]$ via the affine transformation

$$t \equiv t(\tau, t_0, t_f) = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}. \tag{5}$$

In order to discretize the optimal control problem using an *hp* method, the domain $\tau \in [-1, +1]$ is partitioned into a *mesh* consisting of K *mesh intervals* $\mathcal{S}_k = [T_{k-1}, T_k]$, $k = 1, \ldots, K$, where $-1 = T_0 < T_1 < \ldots < T_K = +1$. The mesh intervals have the property that $\bigcup_{k=1}^{K} \mathcal{S}_k = [-1, +1]$. Let $\mathbf{y}^{(k)}(\tau)$ and $\mathbf{u}^{(k)}(\tau)$ be the state and control in $\mathcal{S}_k$. Using the transformation given in Eq. (5), the Bolza optimal control problem of Eqs. (1)–(4) can then rewritten as follows. Minimize the cost functional

$$\begin{aligned}\mathcal{J} = &\mathcal{M}(\mathbf{y}^{(1)}(-1), t_0, \mathbf{y}^{(K)}(+1), t_f) \\ &+ \frac{t_f - t_0}{2}\sum_{k=1}^{K}\int_{T_{k-1}}^{T_k} \mathcal{L}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t)\, d\tau,\end{aligned} \tag{6}$$

subject to the dynamic constraints

$$\frac{d\mathbf{y}^{(k)}(\tau)}{d\tau} - \frac{t_f - t_0}{2}\mathbf{a}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t) = \mathbf{0}, \quad (k = 1, \ldots, K), \tag{7}$$

the path constraints

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t) \leq \mathbf{c}_{\max}, \quad (k = 1, \ldots, K), \tag{8}$$

and the boundary conditions

$$\mathbf{b}_{\min} \leq \mathbf{b}(\mathbf{y}^{(1)}(-1), t_0, \mathbf{y}^{(K)}(+1), t_f) \leq \mathbf{b}_{\max}. \tag{9}$$

Because the state must be continuous at each interior mesh point, it is required that the condition $\mathbf{y}(T_k^-) = \mathbf{y}(T_k^+)$, $(k = 1, \ldots, K-1)$ be satisfied at the interior mesh points $(T_1, \ldots, T_{K-1})$.

IV. Legendre-Gauss-Radau Collocation

The multiple-interval form of the continuous-time Bolza optimal control problem in Section III is discretized using collocation at Legendre-Gauss-Radau (LGR) points [11–13, 17]. In the LGR collocation method, the state of the continuous-time Bolza optimal control problem is approximated in $\mathcal{S}_k$, $k \in [1, \ldots, K]$, as

$$\begin{aligned}\mathbf{y}^{(k)}(\tau) \approx \mathbf{Y}^{(k)}(\tau) &= \sum_{j=1}^{N_k+1} \mathbf{Y}_j^{(k)} \ell_j^{(k)}(\tau) \\ \ell_j^{(k)}(\tau) &= \prod_{\substack{l=1 \\ l\neq j}}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}},\end{aligned} \tag{10}$$

where $\tau \in [-1, +1]$, $\ell_j^{(k)}(\tau)$, $j = 1, \ldots, N_k + 1$, is a basis of Lagrange polynomials, $\left(\tau_1^{(k)}, \ldots, \tau_{N_k}^{(k)}\right)$ are the Legendre-Gauss-Radau (LGR) [11] collocation points in $\mathcal{S}_k = [T_{k-1}, T_k)$, and $\tau_{N_k+1}^{(k)} = T_k$ is a noncollocated point. Differentiating $\mathbf{Y}^{(k)}(\tau)$ in Eq. (10) with respect to τ gives

$$\frac{d\mathbf{Y}^{(k)}(\tau)}{d\tau} = \sum_{j=1}^{N_k+1} \mathbf{Y}_j^{(k)} \frac{d\ell_j^{(k)}(\tau)}{d\tau}. \tag{11}$$

Defining $t_i^{(k)} = t(\tau_i^{(k)}, t_0, t_f)$ using Eq. (5), the dynamics are then approximated at the N_k LGR points in mesh interval $k \in [1, \ldots, K]$ as

$$\begin{gathered} \sum_{j=1}^{N_k+1} D_{ij}^{(k)} \mathbf{Y}_j^{(k)} - \frac{t_f - t_0}{2} \mathbf{a}(\mathbf{Y}_i^{(k)}, \mathbf{U}_i^{(k)}, t_i^{(k)}) = \mathbf{0}, \\ (i = 1, \ldots, N_k), \end{gathered} \tag{12}$$

where $D_{ij}^{(k)} = d\ell_j^{(k)}(\tau_i^{(k)})/d\tau$, $(i = 1, \ldots, N_k)$, $(j = 1, \ldots, N_k + 1)$ are the elements of the $N_k \times (N_k + 1)$ *Legendre-Gauss-Radau differentiation matrix* [11] in mesh interval $\mathcal{S}_k$, $k \in [1, \ldots, K]$. The LGR discretization then leads to the following nonlinear programming problem (NLP). Minimize

$$\begin{aligned} \mathcal{J} \approx & \mathcal{M}(\mathbf{Y}_1^{(1)}, t_0, \mathbf{Y}_{N_K+1}^{(K)}, t_f) \\ & + \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{t_f - t_0}{2} w_j^{(k)} \mathcal{L}(\mathbf{Y}_j^{(k)}, \mathbf{U}_j^{(k)}, t_j^{(k)}) \end{aligned} \tag{13}$$

subject to the collocation constraints of Eq. (12) and the constraints

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{Y}_i^{(k)}, \mathbf{U}_i^{(k)}, t_i^{(k)}) \leq \mathbf{c}_{\max}, \ (i = 1, \ldots, N_k), \tag{14}$$

$$\mathbf{b}_{\min} \leq \mathbf{b}(\mathbf{Y}_1^{(1)}, t_0, \mathbf{Y}_{N_K+1}^{(K)}, t_f) \leq \mathbf{b}_{\max}, \tag{15}$$

$$\mathbf{Y}_{N_k+1}^{(k)} = \mathbf{Y}_1^{(k+1)}, \quad (k = 1, \ldots, K-1) \tag{16}$$

where $N = \sum_{k=1}^{K} N_k$ is the total number of LGR points and Eq. (16) is the continuity condition on the state and is enforced at the interior mesh points $(T_1, \ldots, T_{K-1})$ by treating $\mathbf{Y}_{N_k+1}^{(k)}$ and $\mathbf{Y}_1^{(k+1)}$ as the same variable in the NLP.

Finally, it is noted that the mesh refinement method developed in this paper requires an estimate of the solution error on the current mesh. In this paper the approach for estimating the solution relative error that used in Ref. [17] and compares two approximations to the state (one with higher accuracy than the other). The details of this relative error estimate are beyond the scope of this paper and the reader is referred to Ref. [17] for the details.

V. Problem Formulation

A. Equations of Motion

The equations of motion for a point mass in motion over a spherical non-rotating planet are given as [22]

$$\begin{aligned} \dot{r} &= v \sin\gamma, \\ \dot{v} &= -D - g \sin\gamma, \\ \dot{\theta} &= \frac{v \cos\gamma \sin\psi}{r \cos\phi}, \\ \dot{\phi} &= \frac{v \cos\gamma \cos\psi}{r}, \\ \dot{\gamma} &= \frac{1}{v}\left[L \cos\sigma + \left(\frac{v}{r} - \frac{g}{v}\right)\cos\gamma\right], \\ \dot{\psi} &= \frac{L \sin\sigma}{v \cos\gamma} + \frac{v \cos\gamma \sin\psi \tan\theta}{r}, \end{aligned} \tag{17}$$

where $g = \mu/r^2$. Next, the lift and drag specific forces are defined, respectively, as

$$\begin{aligned} L &= \frac{qC_L S}{m}, \\ D &= \frac{qC_D S}{m}, \end{aligned} \tag{18}$$

where

$$q = \frac{1}{2}\rho v^2 \tag{19}$$

is the dynamic pressure,

$$\rho = \rho_0 e^{-h/H} \tag{20}$$

is the density of the Martian atmosphere, and

$$h = r - R \tag{21}$$

is the altitude over a spherical planet. It is noted that the quantities ρ_0, H, and R, are, respectively, the sea level density, the density scale height, and the radius of the planet (and are all constants). The physical constants used in this study are shown in Table 1.

Table 1 Physical constants.

Parameter	Value
R	3.386×10^6 [m]
μ	4.284×10^{13} [m^3/s^2]
g_0	9.80665 [m/s^2]
S	15.9 [m^2]
C_D	1.45
C_L	0.348
m	3300 [kg]
ρ_0	0.0158 [kg/m^3]
H	9354 [m]
K_q	1.9027×10^{-8} [W/cm^2]
N	0.5
M	3
r_n	0.6 [m]

B. Path Constraints

The path constraints imposed on the aerodynamic pressure, aerodynamic acceleration, and heat rate are defined, respectively, as

$$q \leq q_{\max}, \tag{22}$$

$$A = \frac{\sqrt{L^2 + D^2}}{g_0} < A_{\max}, \tag{23}$$

$$\dot{Q} = Kq \left(\frac{\rho}{r_n}\right)^N v^M < \dot{Q}_{\max}, \tag{24}$$

where N, M, and K_q are constants.

C. Boundary Conditions

The following initial conditions were imposed:

$$\begin{aligned} r(t_0) &= r_0, \\ v(t_0) &= v_0, \\ \theta(t_0) &= \theta_0, \\ \phi(t_0) &= \phi_0, \\ \gamma(t_0) &= \gamma_0, \\ \psi(t_0) &= \psi_0, \end{aligned} \tag{25}$$

where r_0, v_0, θ_0, ϕ_0, γ_0, ψ_0 are specified values. Next, the terminal condition was given as

$$v(t_f) = v_f, \tag{26}$$

where v_f is specified. Next, the following bounds are set for the components of the state during the motion:

$$\begin{aligned} r_{\min} &\leq r \leq r_{\max}, \\ v_{\min} &\leq v \leq v_{\max}, \\ \theta_{\min} &\leq \theta \leq \theta_{\max}, \\ \phi_{\min} &\leq \phi \leq \phi_{\max}, \\ \gamma_{\min} &\leq \gamma \leq \gamma_{\max}, \\ \psi_{\min} &\leq \psi \leq \psi_{\max}, \end{aligned} \tag{27}$$

Additionally, the control, σ, is bounded as follows:

$$\sigma_{\min} \leq \sigma \leq \sigma_{\max}. \tag{28}$$

The numerical values for the boundary conditions are shown in Table 2 while the bounds on all variables and constraints are shown in Table 3.

Table 2 Boundary conditions.

Parameter	Value
r_0	3.5112×10^6 [m]
θ_0	0 [rad]
v_0	6000 [m/s]
ϕ_0	-0.0873 [rad]
γ_0	-0.2007 [rad]
ψ_0	1.6581 [rad]
v_f	540 m/s

D. Mars Entry Optimal Control Problem

The Mars entry optimal control problem is now stated as follows. Maximize the terminal altitude, that is, maximize

$$\max \; h(t_f) \tag{29}$$

subject to the dynamic constraints of Eq. (17), the path constraints of Eqs. (22)–(24), the initial conditions of Eq. (25), the terminal condition of Eq. (26), the state bounds given in Eq. (27), and the control bounds given in Eq. (28).

Table 3 Bounds on variables and constraints.

Parameter	Value	Parameter	Value
$r_{\min}$	3.3862×10^6 [m]	$r_{\max}$	3.5112×10^6[m]
$v_{\min}$	0.1 [m/s]	$v_{\max}$	6000 [m/s]
$\theta_{\min}$	-2π [rad]	$\theta_{\max}$	2π [rad]
$\phi_{\min}$	-1.2217 [rad]	$\phi_{\max}$	1.2217 [rad]
$\gamma_{\min}$	-1.0472 [rad]	$\gamma_{\max}$	0 [rad]
$\psi_{\min}$	-2π [rad]	$\psi_{\max}$	2π [rad]
$\sigma_{\min}$	0.05236 [rad]	$\sigma_{\max}$	2. 0944 [rad]
$q_{\min}$	0 [Pa]	$q_{\max}$	10000 [Pa]
$A_{\min}$	0 [m/s^2]	$A_{\max}$	$5g_0$ [m/s^2]
$Q_{\min}$	0 [W/cm^2]	$Q_{\max}$	70 [W/cm^2]

VI. Reference Optimal Solution and Guidance Strategy

A. Reference Optimal Solution

The reference optimal control problem formulated in Section V was solved using the optimal control software $\mathbb{GPOPS}-\mathbb{II}$. $\mathbb{GPOPS}-\mathbb{II}$ employs adaptive Gaussian quadrature collocation [11–16, 18, 19] This method allows smaller meshes to be used than if an h method, a fixed order method, was utilized, all the while achieving the desired accuracy tolerances. The adaptive method does this by placing a larger number of mesh points where the solution is nonsmooth or experiences rapid changes, and places fewer mesh points where the solution is smooth. The smoothness of the solution was determined by approximating the state using a basis of Legendre polynomials and assessing the decay rate of the polynomial coefficients as a function of the coefficient index [19]. $\mathbb{GPOPS}-\mathbb{II}$ also implements the nonlinear programming problem solver, IPOPT, an interior point method [23]. IPOPT was employed in full Newton mode, and supplies the objective function gradient, Lagrangian Hessian, and constrain Jacobian. Sparse central finite differencing was used to supply IPOPT with the first and second derivatives. When setting up the optimal control problem in $\mathbb{GPOPS}-\mathbb{II}$, the initial conditions, boundary conditions, model parameters, an initial guess, and an initial mesh were used as inputs. A continuous function was created to compute the spacecraft dynamics, state, and path constraints. An objective function was created to define the desired end goal, which in this problem was to maximize the terminal altitude. $\mathbb{GPOPS}-\mathbb{II}$ outputs the optimal solution containing the state matrix and corresponding control matrix.

B. Guidance Strategy

In order to explore the effect of modeling errors and perturbations from the reference solution, the following strategy was employed and follows an approach similar to that described in Ref. [20]. First, it was assumed that the coefficient of drag for the actual vehicle is different from the reference value of C_D. Then, using the reference solution as the starting point, the reference optimal control problem was computed using $\mathbb{GPOPS}-\mathbb{II}$. The flight of the vehicle is then simulated on the time interval $[t_0, t_0 + \Delta T]$ using the perturbed model where ΔT is the guidance cycle time with the MATLAB ordinary differential equation solver ODE113. The state of the vehicle attained from the simulation with the perturbed model is then used as the initial condition for solving the optimal control problem on the remaining horizon (that is, the horizon $t_0 + \Delta T, t_f$]). The process of simulating the flight with the perturbed model is then repeated over each subsequent guidance cycle ΔT and the resulting state at the end of the guidance cycle is using as the initial condition when solving the optimal control problem over a progressively shrinking horizon until the remaining horizon is smaller than ΔT (at which point the last computed control is used to simulate the remainder of the flight). In this manner, the remaining horizon is $t_0 + k\Delta T, t_f$, where $t_0 + k\Delta T$, where K is the total number of guidance cycles. In order to obtain a solution quickly to the optimal control problem, at the start of each guidance cycle the starting mesh used for $\mathbb{GPOPS}-\mathbb{II}$ consists of only that portion of the mesh that corresponds to the remaining horizon. In other words, the portion of the mesh that occurs prior to the time $t_0 + k\Delta T$ is deleted and only that portion of the mesh from $[t_0 + k\Delta T, t_f]$ is used. Using this process, the mesh on the remaining horizon is smaller than the mesh used on the

previous remaining horizon. By using a smaller mesh on each remaining horizon the problem can be solved more quickly as the guidance cycles evolve because the resulting NLP from $\mathbb{GPOPS} - \mathbb{II}$ becomes progressively smaller. it should be noted that all computations were performed on a 2.4 GHz 8-Core Intel Core i9 2019 MacBook Pro with 32 GB 2667 MHz DDR4 running on Mac OS version 11.3.1 Big Sur. The computations were performed by MATLAB Version R2019b.

VII. Results and Discussion

This section explores the results obtained by computing the optimal trajectory and implementing guidance updates as outlined in Section VI. The modeling errors were simulated by varying the coefficient of drag of the entry vehicle when integrating the dynamics to simulate the actual motion of the spacecraft. Multiple values of C_D were used, however only values of $C_D = 1,\ 1.9$ are presented in this paper. When completing the guidance updates presented in Subsection VII.B and Subsection VII.C, a guidance cycle period of 10 seconds was used. The guidance cycle period was chosen such that the computational time was sufficiently smaller than the guidance cycle period. Finally, the effect of the guidance cycle period was studied in Subsection VII.D.

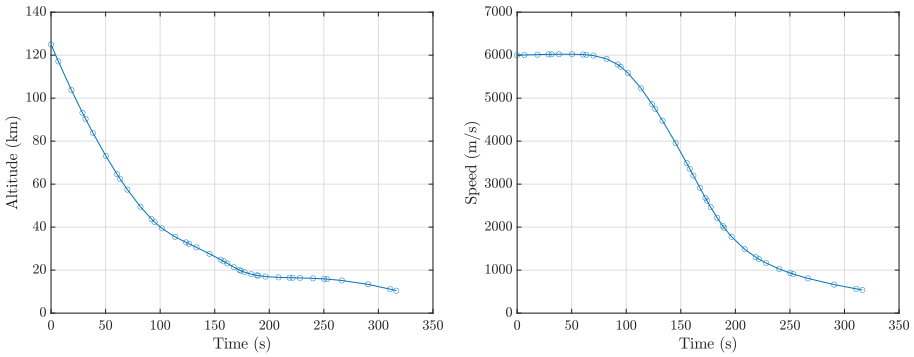
A. Reference Solution

When solving the reference optimal control problem, i.e using a coefficient of drag value of $C_D = 1.45$, the final state of the spacecraft was found to be:

$$\mathbf{y}(t_f) = [10.4 \text{ km} + R \quad 540 \text{ m/s} \quad 0.306 \text{ rad} \quad -0.137 \text{ rad} \quad -0.246 \text{ rad} \quad 2.07 \text{ rad}] \tag{30}$$

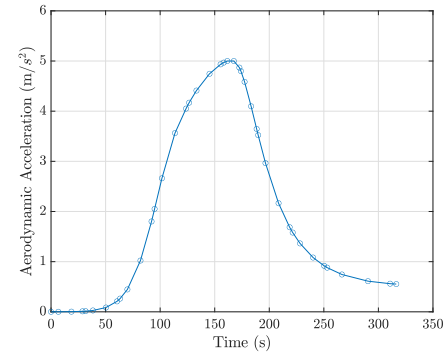
The spacecraft altitude, speed, aerodynamic acceleration, aerodynamic pressure, heat rate and bank angle of the spacecraft versus time for the reference solution are shown in Fig. 1. The following observations can be made from the reference trajectory. First, the vehicle enters the atmosphere at approximately 60 km/s and holds this speed for approximately 100 seconds, at which point the Martian atmospheric density becomes sufficiently large enough to produce enough drag force to lower the velocity. This point occurs at an altitude of approximately 40 km. From Fig. 1(c) it can be seen that the path constraint on the acceleration becomes active roughly 150 seconds into entry. As the vehicle continues its entry phase it can be seen that it's speed continues to decrease until it reaches a value of 540 km/s, at which a parachute would be deployed and the descent phase of the mission would begin. This occurs at a maximized final altitude of 10.4 km.

When observing the control generated by solving the optimal control problem, it can be seen that it roughly follows a bang-bang control pattern, switching between the minimum bank angle, $\sigma = 30°$ and the maximum bank angle $\sigma = 120°$. The minimum bank angle was chosen such that it was large enough to ensure the lift vector would not point vertical,, and the maximum bank angle was chosen such that it was small enough to ensure the vehicle would not be at risk of entering a spin. The bang-bang behavior seen is a result of the spacecraft depleting speed quickly while banking at a value $\sigma = 30°$ and then flipping to a value of $\sigma = 120°$ to ensure the heating rate does not exceed the maximum allowable value.

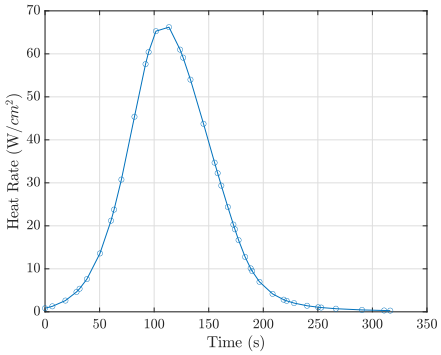


(a) $h(t)$ vs. t.

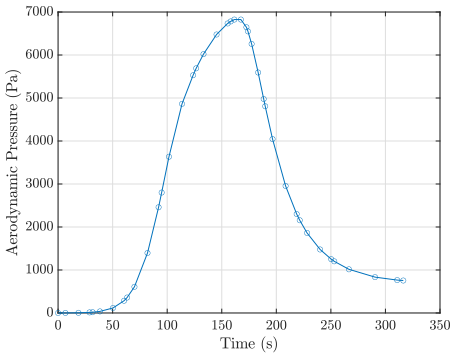
(b) $v(t)$ vs. t.



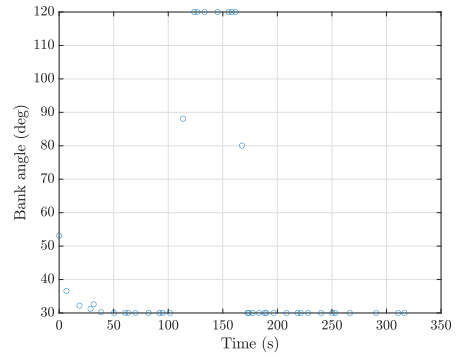
(c) $A(t)$ vs. t.



(d) $\dot{Q}(t)$ vs. t.



(e) $q(t)$ vs. t.



(f) $\sigma(t)$ vs. t.

Fig. 1 Results for the reference solution.

B. Simulation of flight for the case of $C_D = 1$.

Figure 2 shows the spacecraft altitude, speed, aerodynamic acceleration, aerodynamic pressure, and heat rate of the spacecraft versus time for the case of $C_D = 1$ when guidance updates are implemented compared to the reference solution. Additionally, the computation time for each guidance cycle and bank angle was plotted versus time.

The final state for the case of $C_D = 1$ was found to be:

$$\mathbf{y}(t_f) = [8.14\ \text{km} + R \quad 553\ \text{m/s} \quad 0.368\ \text{rad} \quad -0.163\ \text{rad} \quad -0.268\ \text{rad} \quad 2.26\ \text{rad}] \tag{31}$$

The time to reach the final state was 406.5 seconds.

Figure 2(b) shows that during the first 60 seconds approximately, the speed curve for the reference solution and the curve generated through guidance updates overlie each other since the atmospheric density at the beginning of entry is not large enough to cause a drag force sufficiently large enough to slow the vehicle. The curves then diverge, where

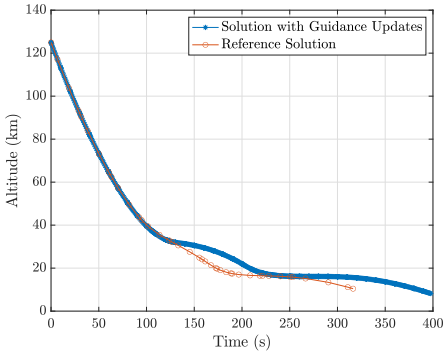
the speed curve for the solution with guidance updates decreases at a slower rate than reference solution curve. The coefficient of drag in this case was smaller than the reference solution and as a result, the drag force on the vehicle will be lower and the vehicle will take longer to slow down to its terminal velocity and will reach a lower final altitude. It is also noted that the final speed does not reach the desired terminal speed.

Additionally, when simulating a modeling error it can be seen in Fig. 2(d) that the heating rate exceeds the maximum allowable value, reaching a maximum value of 77.19 W/cm^2. It can be seen from Fig. 2(f) that when the path constraint is active, the computational time to complete the update spike (the tenth cycle). Larger computational times decrease the effectiveness of completing guidance updates.

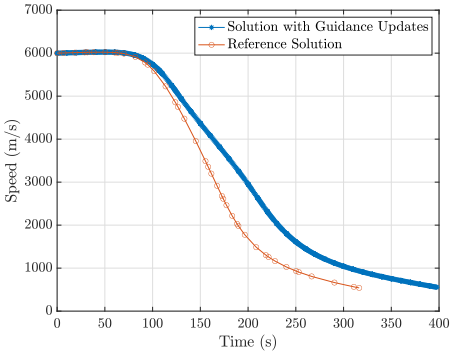
The effectiveness of guidance updates can be observed by solving the optimal control problem with the modeling error while not completing any guidance updates. For the case of $C_D = 1$, when the control generated from reference solution is integrated from the initial to the final time, the final state produced was computed to be:

$$\mathbf{y}(t_f) = [24.1\ \text{km} + R \quad 1110\ \text{m/s} \quad 0.333\ \text{rad} \quad -0.150\ \text{rad} \quad -0.0379\ \text{rad} \quad 2.08\ \text{rad}] \tag{32}$$

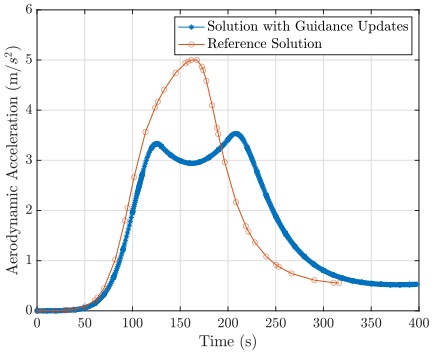
Without guidance updates the vehicle would be traveling at more than double the desired terminal speed at the time the reference solution would predict the parachute to be deployed.



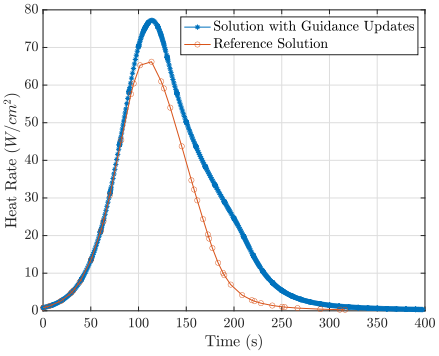
(a) $h(t)$ vs. t.



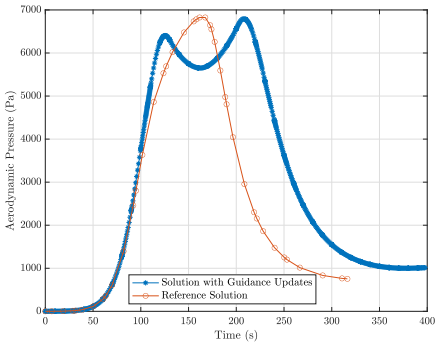
(b) $v(t)$ vs. t.



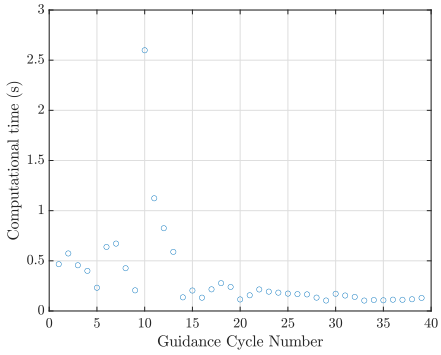
(c) $A(t)$ vs. t.



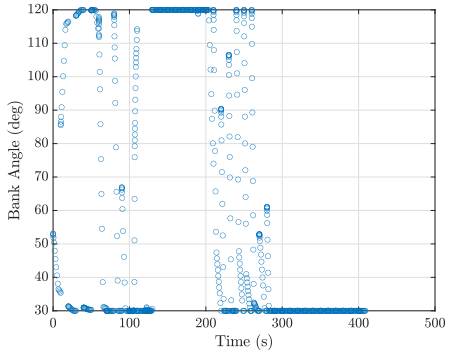
(d) $\dot{Q}(t)$ vs. t.



(e) $q(t)$ vs. t.



(f) Computation time vs. n.



(g) $\sigma(t)$ vs. t.

Fig. 2 Results for the case of $C_D = 1$.

C. Simulation of flight for the case of $C_D = 1.9$

The following figures show the height above the Mars surface, speed, aerodynamic acceleration, aerodynamic pressure, and heat rate of the spacecraft versus time for the case of $C_D = 1.9$ with the implementation of guidance updates compared to the reference solution. Additionally, the computation time for each guidance cycle and the bank angle were plotted.

The final state for the case of $C_D = 1.9$ was found to be:

$$\mathbf{y}(t_f) = [11.9\text{ km} + R \quad 532\text{ m/s} \quad 0.284\text{ rad} \quad -0.127\text{ rad} \quad -0.235\text{ rad} \quad 1.97\text{ rad}] \tag{33}$$

The time to reach the final state was 280 seconds.

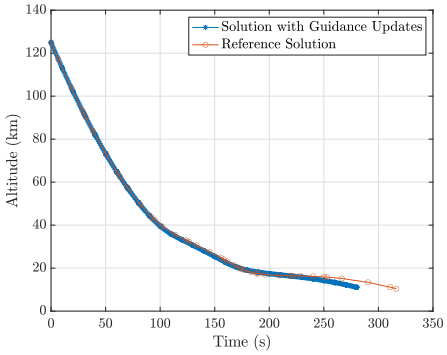
It can be seen from Fig. 3 that the curves of the solution computed with guidance updates follow the reference solution curves more closely then in the case of $C_D = 1$. Additionally the vehicle achieves its desired terminal velocity in a shorter amount of time than the reference solution, and at a higher terminal altitude. However, due to the increased drag force, the maximum allowable aerodynamic acceleration is exceeded, reaching a maximum value of $5.19 g_0$ [m/s^2].

While it may seem that the performance of the vehicle compared to the case of $C_D = 1$, it should be noted that at the very last guidance cycle, the computation time has a significant spike. This could imply that the optimal control problem is becoming infeasible.

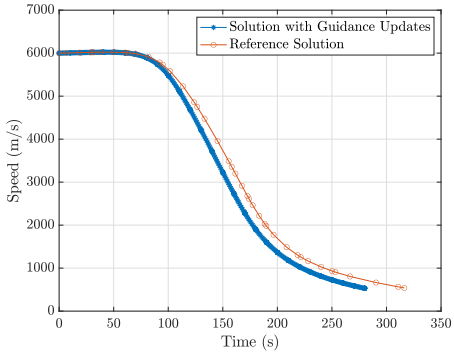
The effectiveness of the guidance updates can be demonstrated by computing the state of the vehicle using the control generated from the reference solution and integrating the dynamics including the modeling error. The final state without guidance updates was computed to be:

$$\mathbf{y}(t_f) = [5.04\text{ km} + R \quad 350\text{ m/s} \quad 0.291\text{ rad} \quad -0.130\text{ rad} \quad -0.463\text{ rad} \quad 2.04\text{ rad}] \tag{34}$$

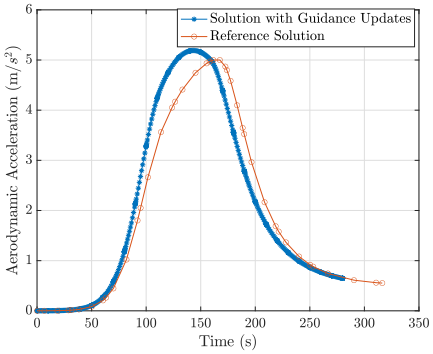
Without guidance updates the vehicle would deploy the parachute at a speed of 350 m/s, leading to a non-optimal terminal altitude.



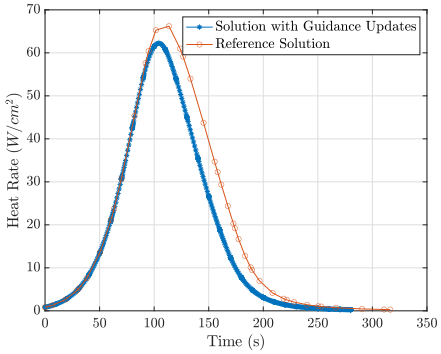
(a) $h(t)$ vs. t.



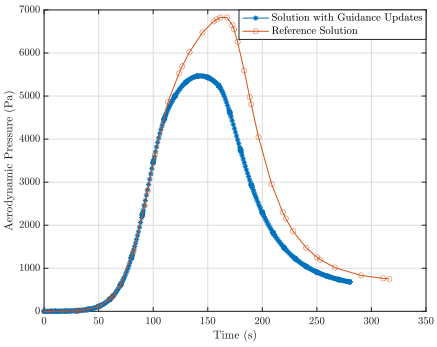
(b) $v(t)$ vs. t.



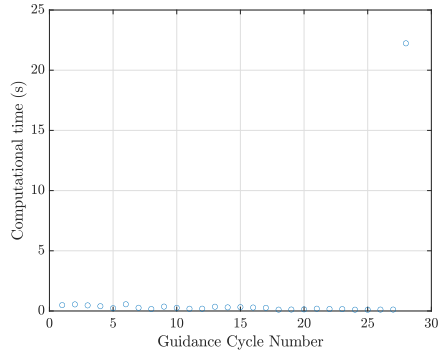
(c) $A(t)$ vs. t.



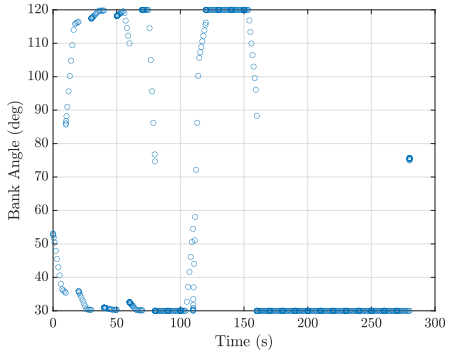
(d) $\dot{Q}(t)$ vs. t.



(e) $q(t)$ vs. t.



(f) Computation time vs. n.



(g) $\sigma(t)$ vs. t.

Fig. 3 Results for the case of $C_D = 1.9$.

D. Comparison of guidance cycle periods

The final altitude and speed were plotted as a function of guidance cycle period in the case of C_D = 1 and the results were compared to the values achieved in the reference solution, as well as the solution with the modeling error and no guidance updates. (Fig. 4 and 5).

The shortest guidance cycle period, every 2.5 seconds, generated the closest final speed to the desired final speed, reaching a terminal speed of 542 m/s. While a guidance cycle of 2.5 seconds allows the terminal speed to reach the necessary speed to deploy the parachute, it came with the price of high computational times. The maximum computational times for this guidance cycle exceed the actual guidance cycle period. The same behavior held for a guidance cycle period of 5 seconds. In this case the terminal speed reached a value of 445 m/s however the maximum computational times still exceeded 5 seconds. The guidance cycle period of 10 seconds produced a terminal speed of 553 m/s. The computational times for this guidance cycle period remained significantly below 10 seconds, which is why a guidance cycle period of 10 seconds was chosen for the earlier analysis presented in Subsection VII.B and Subsection VII.C.

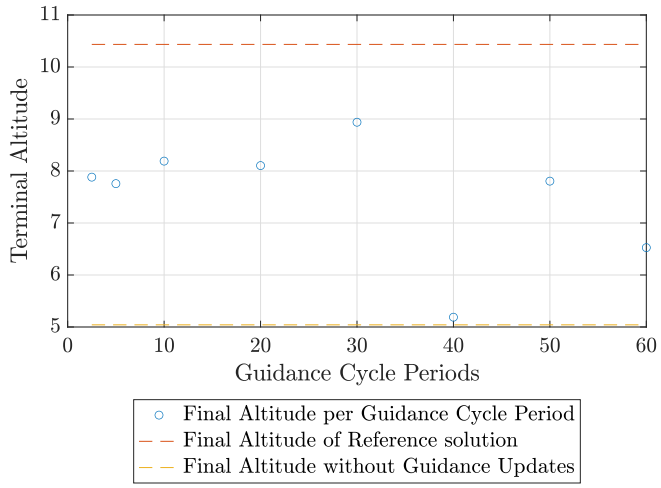


Fig. 4 Altitude versus guidance cycle period. The dashed lines show the final height for the reference solution and the solution obtained by introducing the modeling error without guidance updates. Results were obtained for C_D = 1

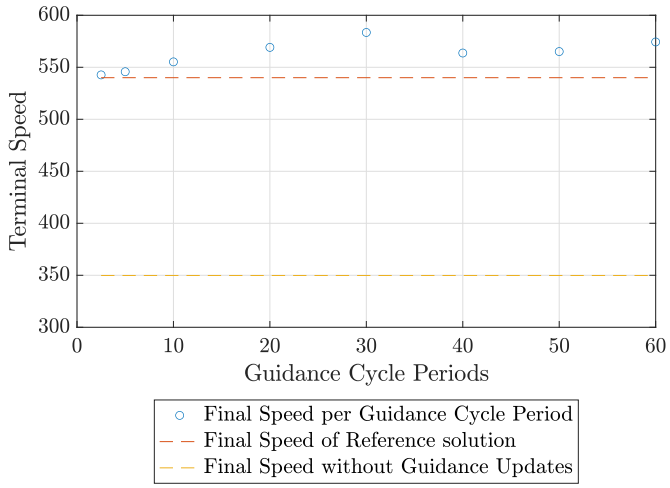


Fig. 5 Speed versus guidance cycle period. The dashed lines show the final speed for the reference solution and the solution obtained by introducing the modeling error without guidance updates. Results were obtained for C_D = 1

VIII. Conclusions

A computational approach for optimal trajectory generation and guidance of a Mars entry vehicle has been presented. The goal was to maximize the terminal altitude of the vehicle when a desired terminal speed is achieved. Additionally, guidance updates were implemented by introducing a modeling error into the problem and resolving the optimal control problem at given guidance cycle periods. It was found that in reference solution using a coefficient of drag of C_D= 1.45, that the optimal trajectory produced a maximum terminal altitude of 10.43 km. Modeling errors were generated by simulating the dynamics of the vehicle with off-nominal coefficient of drag values of 1 and 1.9. Without introducing guidance updates, integrating the dynamics for these off-nominal coefficients of drag using the control generated from the reference solution led to terminal altitudes of 24.1 km and 5.04 km, respectively, corresponding, respectively, to terminal speeds of 1110 m/s and 350 m/s. When implementing guidance updates every 10 seconds, the maximum terminal altitudes computed were 8.14 km and 11.1 km corresponding to terminal speeds of 553 m/s and 532 m/s, respectively.

IX. Acknowledgments

The authors gratefully acknowledge support for this research from the U.S. National Science Foundation under grant CMMI-2031213 and from the U.S. Air Force Research Laboratory under contract FA8651-21-F-1041.

References

[1] Zarchan, P., *Tactical and Strategic Missile Guidance*, 6th ed., Vol. 239, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 2012.

[2] Liu, X., Shen, Z., and Lu, P., "Closed-Loop Optimization of Guidance Gain for Constrained Impact," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 453–460. https://doi.org/10.2514/1.G000323.

[3] Kim, M., and Grider, K. V., "Terminal Guidance for Impact Attitude Angle Constrained Flight Trajectories," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-9, 1973, pp. 852–859. https://doi.org/10.1109/TAES.1973.309659. https://doi.org/10.1109/TAES.1973.309659.

[4] Farooq, A., and Limebeer, D. J. N., "Optimal Trajectory Regulation for Radar Imaging Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1076–1092. https://doi.org/10.2514/1.31441.

[5] Shaferman, V., and Shima, T., "Linear Quadratic Guidance Laws for Imposing a Terminal Intercept Angle," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1400–1412. https://doi.org/10.2514/1.32836.

[6] Jardin, M. R., and Jr., A. E. B., "Neighboring Optimal Aircraft Guidance in Winds," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, 2001, pp. 710–715. https://doi.org/10.2514/2.4798.

[7] Mayne, D. Q., "Model Predictive Control: Recent Developments and Future Promise," *Automatics*, Vol. 50, No. 12, 2014, pp. 2967–2986. https://doi.org/10.1016/j.automatica.2014.10.128.

[8] Rawlings, J. B., Mayne, D. Q., and Diehl, M. M., *Model Predictive Control: Theory, Computation, and Design*, 2nd ed., Nob Hill Publishing, 2017.

[9] Chen, H., and Alllgower, F., "A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability," *Automatica*, Vol. 34, No. 10, 1998, pp. 1205 – 1217. https://doi.org/10.1016/S0005--1098(98)00073--9. https://doi.org/https://doi.org/10.1016/S0005-1098(98)00073-9. URL http://www.sciencedirect.com/science/article/pii/S0005109898000739.

[10] Yu, M., Griffith, D. W., and Biegler, L. T., *Nonlinear Programming Formulations for Nonlinear and Economic Model Predictive Control*, Springer International Publishing, Cham, 2019, pp. 465–489. https://doi.org/10.1007/978-3-319-77489-3_20. URL https://doi.org/10.1007/978-3-319-77489-3_20.

[11] Garg, D., Patterson, M. A., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods," *Automatica*, Vol. 46, No. 11, 2010, pp. 1843–1851. DOI: 10.1016/j.automatica.2010.06.048.

[12] Garg, D., Hager, W. W., and Rao, A. V., "Pseudospectral Methods for Solving Infinite-Horizon Optimal Control Problems," *Automatica*, Vol. 47, No. 4, 2011, pp. 829–837. DOI: 10.1016/j.automatica.2011.01.085.

[13] Garg, D., Patterson, M. A., Darby, C. L., Francolin, C., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems via a Radau Pseudospectral Method," *Computational Optimization and Applications*, Vol. 49, No. 2, 2011, pp. 335–358. DOI: 10.1007/s10589–00–09291–0.

[14] Darby, C. L., Hager, W. W., and Rao, A. V., "An hp–Adaptive Pseudospectral method for Solving Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 32, No. 4, 2010, pp. 476–502. https://doi.org/10.1002/oca.957, URL https://doi.org/10.1002/oca.957.

[15] Darby, C. L., Garg, D., and Rao, A. V., "Costate Estimation using Multiple-Interval Pseudospectral Methods," *Journal of Spacecraft and Rockets*, Vol. 48, No. 5, 2011, pp. 856–866. https://doi.org/10.2514/1.a32040, URL https://doi.org/10.2514/1.a32040.

[16] Patterson, M. A., and Rao, A. V., "GPOPS-II," *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, 2014, pp. 1–37. https://doi.org/10.1145/2558904, URL https://doi.org/10.1145/2558904.

[17] Patterson, M. A., Hager, W. W., and Rao, A. V., "A ph mesh refinement method for optimal control," *Optimal Control Applications and Methods*, Vol. 36, No. 4, 2015, pp. 398–421. https://doi.org/10.1002/oca.2114, URL https://doi.org/10.1002/oca.2114.

[18] Liu, F., Hager, W. W., and Rao, A. V., "Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction," *Journal of the Franklin Institute*, Vol. 352, No. 10, 2015, pp. 4081–4106. https://doi.org/10.1016/j.jfranklin.2015.05.028, URL https://doi.org/10.1016/j.jfranklin.2015.05.028.

[19] Liu, F., Hager, W. W., and Rao, A. V., "Adaptive Mesh Refinement Method for Optimal Control Using Decay Rates of Legendre Polynomial Coefficients," *IEEE Transactions on Control Systems Technology*, Vol. 26, No. 4, 2018, pp. 1475–1483. https://doi.org/10.1109/tcst.2017.2702122, URL https://doi.org/10.1109/tcst.2017.2702122.

[20] Dennis, M. E., Hager, W. W., and Rao, A. V., "Computational Method for Optimal Guidance and Control Using Adaptive Gaussian Quadrature Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 9, 2019, pp. 2026–2041. https://doi.org/10.2514/1.G003943, URL https://arc.aiaa.org/doi/10.2514/1.G003943.

[21] Zheng, Y., H.Cui, and Y.Ai, "Indirect Trajectory Optimization for Mars Entry with Maximum Terminal Altitude," *Journal of Spacecraft and Rockets*, Vol. 54, No. 5, 2017, pp. 1068–1079.

[22] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM, 2010.

[23] Biegler, L. T., and Zavala, V. M., "Large-Scale Nonlinear Programming Using IPOPT: An Integrating Framework for Enterprise-Wide Optimization," *Computers and Chemical Engineering*, Vol. 33, No. 3, 2008, pp. 575–582.