Low-Thrust, High-Accuracy Trajectory Optimization

I. Michael Ross,* Qi Gong,† and Pooya Sekhavat‡
Naval Postgraduate School, Monterey, California 93943

Multirevolution, very low-thrust trajectory optimization problems have long been considered difficult problems due to their large time scales and high-frequency responses. By relating this difficulty to the well-known problem of aliasing in information theory, an antialiasing trajectory optimization method is developed. The method is based on Bellman's principle of optimality and is extremely simple to implement. Appropriate technical conditions are derived for generating candidate optimal solutions to a high accuracy. The proposed method is capable of detecting suboptimality by way of three simple tests. These tests are used for verifying the optimality of a candidate solution without the need for computing costates or other covectors that are necessary in the Pontryagin framework. The tests are universal in the sense that they can be used in conjunction with any numerical method whether or not antialiasing is sought. Several low-thrust example problems are solved to illustrate the proposed ideas. It is shown that the antialiased solutions are, in fact, closed-loop solutions; hence, optimal feedback controls are obtained without recourse to the complexities of the Hamilton–Jacobi theory. Because the proposed method is easy to implement, it can be coded on an onboard computer for practical space guidance.

I. Introduction

CONTINUOUS-THRUST trajectory optimization problems have served as one of the motivating problems for optimal control theory since its inception [1–4]. The classic problem posed by Moyer and Pinkham [2] is widely discussed in textbooks [1,3,4] and research articles [5–7]. When the continuity of thrust is removed from such problems, the results can be quite dramatic as illustrated in Fig. 1. This trajectory was obtained using recent advances in optimal control techniques and is extensively discussed in [8]. In canonical units, the problem illustrated in Fig. 1 corresponds to doubling the semimajor axis ($a_0 = 1$, $a_f = 2$), doubling the eccentricity ($e_0 = 0.1$, $e_f = 0.2$), and rotating the line of apsides by 1 rad. Note that the extremal thrust steering program for minimizing fuel is not tangential over a significant portion of the trajectory. Furthermore, the last burn is a singular control as demonstrated in Fig. 2 by the vanishing of the switching function. Although such finite-thrust problems can be solved quite readily nowadays, it has long been recognized [9–11] that as the thrust authority is reduced, new problems emerge. These well-known challenges chiefly arise as a result of a long flight time measured in terms of the number of orbital revolutions. Consequently, such problems are distinguished from finite-thrust problems as low-thrust problems although the boundary between finite thrust and low thrust is not altogether sharp.

Although ad hoc techniques may circumvent some of the low-thrust challenges, it is not quite clear if the solutions generated from such methods are verifiably optimal. As detailed in [8], the engineering feasibility of a space mission is not dictated by trajectory generation, but by optimality. This is because fuel in space is extraordinarily expensive as the cost of a propellant is driven by the routine of space operations, or the lack of it, and not the chemical composition of the fuel. In an effort to circumvent ad hoc techniques to efficiently solve emerging problems in finite- and low-thrust trajectory optimization, NASA brought together leading experts in

the field to exchange ideas over several workshops. These workshops, held over 2003–2006, further clarified the scope of the problems, and ongoing efforts to address them are described in [12].

From a practical point of view, the goal is to quickly obtain verifiably optimal or near-optimal solutions to finite- and low-thrust problems so that alternative mission concepts can be analyzed
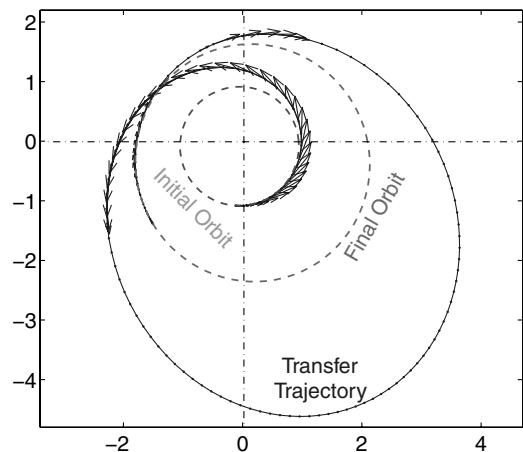

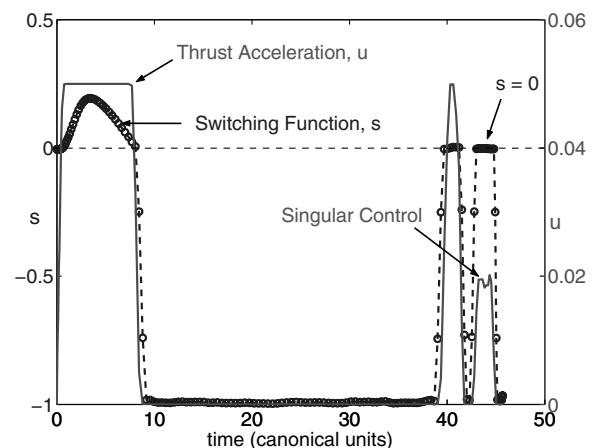Fig. 1 A benchmark minimum-fuel finite-thrust orbit transfer problem.

Fig. 2 Extremal thrust acceleration (control) program $t \mapsto u$ and the corresponding switching function $t \mapsto s$ for the trajectory shown in Fig. 1.

*Professor, Department of Mechanical and Astronautical Engineering; imross@nps.edu. Associate Fellow AIAA.
†Research Associate, Department of Mechanical and Astronautical Engineering; qgong@nps.edu.
‡Research Scientist, Department of Mechanical and Astronautical Engineering; psekhava@nps.edu.

efficiently. The crux of the problem can be described as follows: The difficulties in solving space trajectory optimization problems are the difficulties associated with solving optimal control problems. Thus, the Hamilton–Jacobi–Bellman framework, or dynamic programming, is immediately ruled out as a viable method [8,13]. The Pontryagin framework, frequently called an indirect method in computational techniques [14,15], has substantial detractions that chiefly arise from the symplectic structure of the Hamiltonian system [3,13]. The neo–Bernoulli–Euler framework [16–18], variations of which are called direct methods in computational techniques, offers the best chance for solving optimal control problems [13,15], particularly through an application of the Covector Mapping Principle [7,13,16,18,19]. Among direct methods, direct collocation methods are industry-standard techniques and are available through powerful software packages such as OTIS [20] and SOCS [21]. When traditional direct collocation methods are applied to low-thrust problems, the resulting nonlinear programming (NLP) problem grows substantially large to meet reasonable accuracy requirements [22]. Consequently, large-scale NLP solvers are critical to the success of this approach. As a matter of fact, optimal control problems have been one of the major motivations for research in large-scale NLP methods [23]. An excellent overview of the challenges in solving such problems is described by Betts [14,15]. Other relevant issues, specific to space trajectory optimization problems, are discussed in [8].

In this paper, we introduce a simple technique for solving low-thrust trajectory optimization problems. In exploiting Bellman's principle of optimality, our proposed method rapidly generates high-accuracy solutions to low-thrust problems. Our approach does not require large-scale NLP solvers and is based on interpreting a low-order discrete solution to the problem as an "alias" to a high-order solution. As these repeatedly calculated solutions can be obtained within fractions of a second to a few seconds, even when implemented within a MATLAB[TM] environment running on legacy computer hardware, they are real-time solutions. Thus, feedback guidance is obtained quite simply by real-time computation and updates to the optimal solution [8,24,25]. A brief introduction to the method has also been presented in [26].

It is important to note that while our proposed method does not require large-scale NLP solvers for a single agent (i.e., vehicle) problem, it does not mean that large-scale NLP solvers are unnecessary for optimal control. On the contrary, the ubiquity of large-scale solvers implies that our method can be easily applied to solve multi-agent problems such as formation design and control, autonomous collision avoidance, and other emerging problems as illustrated in [8,27–30].

II. Aliasing and Optimal Control

Consider the discretization (sampling) of two continuous functions (signals) shown in Fig. 3. It is clear that both functions are indistinguishable (or *aliases* of one another) when discretized. This well-known phenomenon in digital information theory causes high-frequency signals to be aliased with low-frequency signals resulting in an incorrect reconstruction of the analog signal. If the high-frequency signal is noise and the low-frequency signal is genuine information, a simple remedy for aliasing is low-pass filtering. If high-frequency components are necessary for signal reconstruction, then the well-known sampling theorem in information theory (attributed to Nyquist, Shannon, Whittaker, and others [31]) states the following:

Theorem 1 (Nyquist–Shannon): For reconstructing a band-limited analog signal, the sampling frequency of the digitizer must be greater than twice the signal bandwidth.

In applying this theorem to low-thrust multirevolution orbit transfer problems, it appears that a sufficiently large grid is needed to capture the "high-frequency" information of orbital revolutions. Deferring a discussion of the specifics to later sections, let us first illustrate this "signal-processing" notion as it applies to optimal control. To this end, consider the simple optimal control test problem:
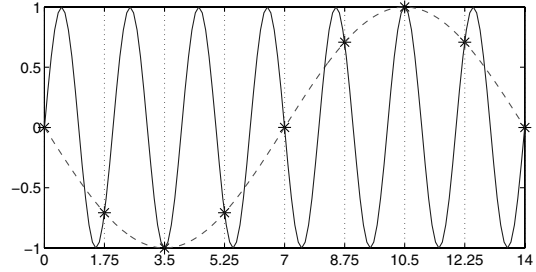

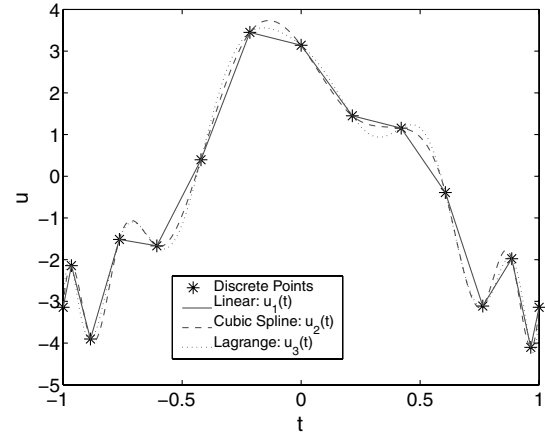Fig. 3   Aliasing in discretization; stars denote the discretization points.

Fig. 4   Continuous-time reconstructions of a control trajectory using various interpolation schemes.

$$(T) \begin{cases} & x \in \mathbb{R}, \ u \in \mathbb{R} \\ \text{Minimize} & J[x(\cdot), u(\cdot)] = \int_{-1}^{1} (x(t) - \sin(\pi t))^2 \, dt \\ \text{Subject to} & \dot{x}(t) = \sin(4\pi x(t)) + u(t) \\ & x(-1) = 0 \end{cases}$$

A 15-point discrete-time solution for a candidate optimal control $u(t_i)$, $i = 1, \ldots, 15$ is shown in Fig. 4. The details of how this solution was obtained are irrelevant to this part of the discussion and are hence deferred. Also shown in Fig. 4 are three suggestions for reconstructing the continuous-time control, $[-1, 1] \ni t \to u$. In principle, one could argue that all three reconstructions are equally valid prima facie. Arguably, an ad hoc approach to picking the best reconstruction is to numerically propagate the differential equation,

$$\dot{x} = \sin(4\pi x) + u(t), \qquad x(-1) = 0$$

for all three reconstructions, $u(t) = u_1(t)$, $u_2(t)$, and $u_3(t)$, and then compare the propagated trajectories with the discrete-time trajectory at all discrete points t_i, $i = 1, \ldots, n$, and/or just the final time. This concept is illustrated in Fig. 5. It is clear from Fig. 5 that all three interpolants produce the same qualitative solution. However, none of the continuous-time reconstructions match the discrete-time trajectory. Assuming that this matching is required, one could now argue that 15 points were insufficient to meet accuracy requirements and thus seek to resolve the problem for a higher number of points. This is the concept upon which standard mesh refinement principles are based [15,32]. This approach has the potential to generate a very large-scale optimization problem.

Now, it can be easily verified that the exact solution to problem T has the following analytical form:

$$x^*(t) = \sin(\pi t) \qquad u^*(t) = \pi \cos(\pi t) - \sin(4\pi x^*(t))$$

A plot of the exact optimal control is shown in Fig. 6 along with the discrete-time solution previously depicted in Fig. 4. It is apparent from this figure that the 15-point discrete-time controller lies on the analytical solution. That is, pointwise convergence was indeed
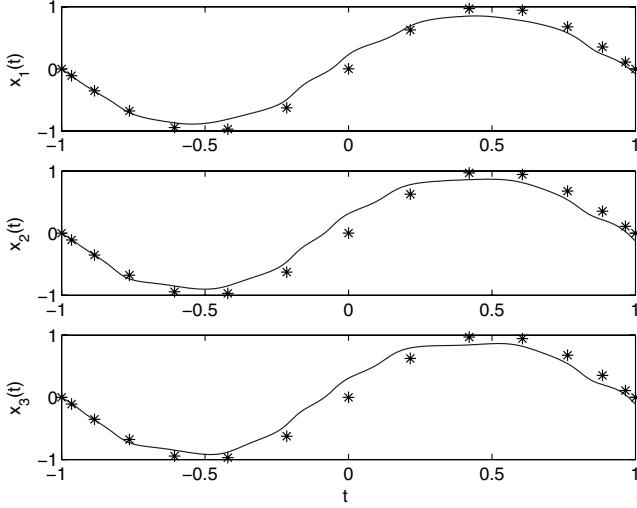
Fig. 5 Discrete-time states (stars) versus numerically propagated trajectories for the continuous-time controls of Fig. 4.
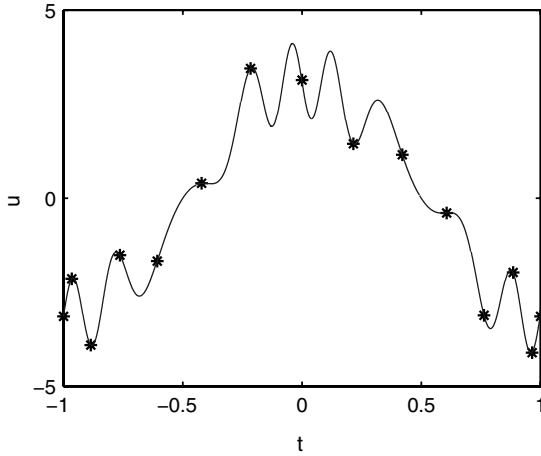

Fig. 6 Comparing a 15-point discrete-time control solution (stars) with the exact solution (solid line).

obtained with just 15 points. In comparing Fig. 6 with Fig. 4, it is clear that none of the reconstruction methods generated the true solution despite the fact that the solution had converged strongly (pointwise). This phenomenon is exactly the same as the aliasing problem illustrated in Fig. 3. Consequently, in taking a cue from the sampling theorem, one can conclude that a larger number of points are necessary to obtain lower "granularity." The 60-point solution, shown in Fig. 7, illustrates how the high-frequency signal is well represented. This problem illustrates, by way of a knowledge of the exact solution, a connection between the sampling theorem and a generic mesh refinement method that calls for increasing the number of sample points. Although 60 "nodes" are by no means large scale, this example illustrates the main point that the 15-node solution is an alias of the 60-node solution, and that aliasing has indeed occurred in an optimal control problem.

III. Antialiasing via Bellman's Principle

A simple remedy for aliasing in optimal control is to refine the mesh along the lines implied by the Nyquist–Shannon sampling theorem. Because problem T is scalar in its state and control variables, the size of the optimization problem is small even when the mesh size is increased from 15 to 60 nodes. For more realistic problems, one can easily conceive mesh sizes of several hundred to thousands of nodes. This generates a large-scale nonlinear programming problem. Large-scale NLPs are now quite common, thanks to a substantial progress in numerical techniques for
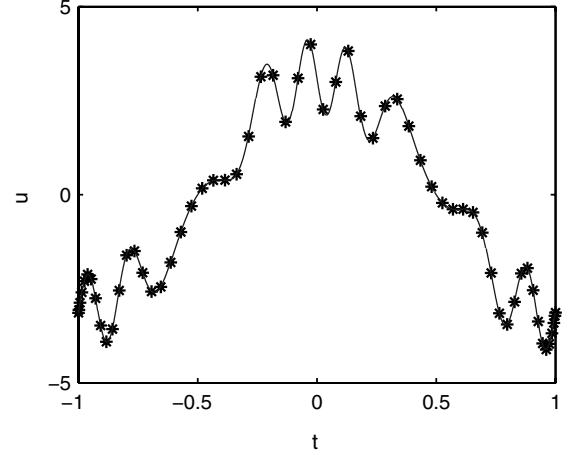

Fig. 7 Mesh-refined 60-node discrete-time control solution.

optimization, and the continuing exponential growth of computer processing capabilities. For example, Gill et al [23] solved over a thousand example problems all with up to 40,000 variables and constraints, while Betts and Erb [22] solved a complex trajectory optimization problem with over 200,000 variables and about 150,000 constraints. Ferris and Munson [33] demonstrate techniques for solving 10×10^6 variable quadratic programming problems. As impressive as these numbers are, it is possible to reduce the size of a discrete optimal control problem by incorporating Bellman's principle of optimality as part of an antialiasing technique. Bellman's principle can be stated as follows:

Principle of optimality: Given an optimal trajectory from a point A to a point B, then the trajectory to point B from a point C lying on the optimal trajectory is also optimal.

Based on the principle of optimality, we propose the following antialiasing Bellman (a^2B) algorithm:

a^2B algorithm:

1) Solve the problem for a low number of nodes n. This generates a discrete-time solution, $\{\mathbf{x}_i, \mathbf{u}_i\}_{i=0}^n$ corresponding to discrete times $\{t_i\}_{i=0}^n$.

2) Partition the time interval $[t_0, t_n]$ into N_B Bellman segments, $t_0 < t^1 < \cdots < t^{N_B} = t_n$. These segments need not be uniformly spaced.

3) Propagate the differential equation from t_0 to t^1 using $\mathbf{x}_0$ as the initial condition and any method of continuous-time reconstruction of the controls, $\mathbf{u}^1(t)$, $t \in [t_0, t^1]$ based on $\{\mathbf{u}_i\}_{i=0}^n$. That is, solve the initial value problem,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}^1(t)), \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \qquad (1)$$

This step generates a continuous-time trajectory, $\mathbf{x}^1(t)$, $t \in [t_0, t^1]$. This propagation is done numerically via some high-precision propagator, say the standard 4/5 Runge–Kutta method.

4) Set $\mathbf{x}_0 = \mathbf{x}^1(t^1)$ and $t_0 = t^1$ and go to step 1; that is, set a new initial condition as the value of the integrated state at the end of the period $[t_0, t^1]$ and solve the problem again for n (which continues to be low). This generates a new sequence $\{\mathbf{x}_i, \mathbf{u}_i\}_{i=0}^n$ corresponding to new discrete times $\{t_i\}_{i=0}^n$, etc.

5) The algorithm stops at the N_Bth sequence when the final conditions are met. The candidate optimal trajectory is given by the Bellman chain $\{\mathbf{x}^1(t), t \in [t_0, t^1]; \mathbf{x}^2(t), t \in [t^1, t^2]; \cdots \mathbf{x}^{N_B}(t), t \in [t^{N_B-1}, t^{N_B}]\} := \mathbf{x}_B(t)$, $t \in [t_0, t_f]$. Similarly, the corresponding controls are given by $\{\mathbf{u}^1(t), t \in [t_0, t^1]; \mathbf{u}^2(t), t \in [t^1, t^2]; \cdots \mathbf{u}^{N_B}(t), t \in [t^{N_B-1}, t^{N_B}]\} := \mathbf{u}_B(t)$, $t \in [t_0, t_f]$.

Remark: It is clear that the practical requirements for our algorithm are an integrator and a legitimate trajectory optimization solver for some small n. Later, we describe several variants of the algorithm, one of which does not even require an integrator. Thus, the only true requirement is an ability to solve discrete optimal control problems for some n. There are a plethora of methods and software available to solve this problem; see [12]. In principle, any of these methods can be

used. In this paper, we have used a particular version of pseudospectral methods [6,34,35]. All of the computations reported in this paper were obtained from the MATLAB-based software package DIDO [35] and executed on a Pentium M, 1.6 GHz laptop with 256 MB of RAM. Depending upon the problem, the computation time for the problems reported in this paper range from fractions of a second to a few seconds. By any metric, this is real-time computation for most orbital applications and hence the antialiased controls are, in fact, closed-loop controls. Additional details on such closed-loop controls are provided in [24,25].

A. Illustrating the Performance of the Algorithm

We now show, by way of the test problem (problem T), how our algorithm generates a high-accuracy trajectory for a low-node discretized system. Taking $n = 15$ as before, we apply our a^2B algorithm to problem T. The result is shown in Fig. 8. Also plotted in the figure is the analytical optimal solution. It is clear that the a^2B solution is almost indistinguishable from the analytical solution. This point is further clarified by the error plots depicted in Fig. 9. These errors are the pointwise differences between the a^2B-computed solution and the analytical solution.

B. Variations of the Algorithm

Several variations of the a^2B algorithm are possible, each with different practical consequences. For example, a simpler algorithm can be designed by picking the ends of the Bellman time segments to be exactly at the node points, $t^k \in \{t_i\}_{i=0}^n$, thus eliminating step 3 of the algorithm. Although this is conceptually simpler, it can be viewed as the special case of the more general a^2B algorithm. In this case, $t^k \in \{t_i\}_{i=0}^n$; for example, say $t^1 = t_m$, $m \ll n$. Therefore, instead of propagating the differential equation, we simply take $\{x_i, u_i\}_{i=0}^m$ as the solution over the Bellman segment and proceed. Our experience with this variant of the algorithm suggests that using the a^2B algorithm in conjunction with a propagator allows for selecting a lower number of nodes n than otherwise. Furthermore, because results from numerical propagators are commonly interpreted as the

"truth," their inclusion as part of our algorithm enhances its practical value as it automatically demonstrates that the "Bellman-sequenced trajectory" is at least feasible from an engineering viewpoint.

IV.   Theoretical Foundations

The main justification for the a^2B algorithm comes from the notion of pointwise convergence over a coarse grid. More specifically, as was shown by the case study of problem T, a fine mesh is a sufficient but not a necessary condition for pointwise convergence. Therefore, it is theoretically possible to achieve convergence without using a fine mesh. Hence, we seek convergence theorems for a theoretical justification of our algorithm, particularly because we do not know the exact solution for a general optimal control problem. It is well recognized [18,36,37] that convergence theorems are one of the hardest theorems to prove. This is because discrete solutions may converge to the wrong answer [18] whereas nonconvergent methods may converge to the right answer [37]. Nonetheless, over the last few years, there has been substantial progress in the development of convergence theorems. Because these theorems form the cornerstone of the legitimacy of the a^2B algorithm, we briefly summarize the results before developing a selection criterion for n, the minimum number of nodes necessary for pointwise convergence.

The most popular discrete methods for control are Euler, Hermite–Simpson, and pseudospectral. Euler and Hermite–Simpson methods are Runge–Kutta (RK) methods with RK coefficients given by the Butcher matrices [15]:

Euler:

$$A = 0, \qquad b = 1$$

Hermite–Simpson:

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 5/24 & 1/3 & -1/24 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}, \qquad b = \begin{pmatrix} 1/6 \\ 2/3 \\ 1/6 \end{pmatrix}$$
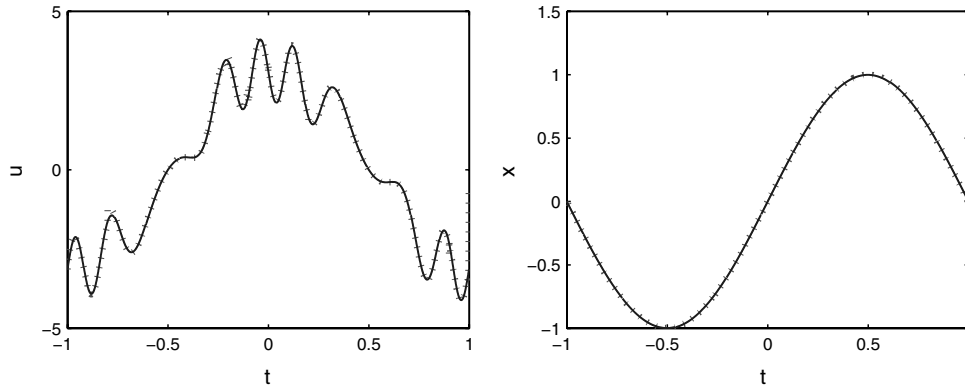

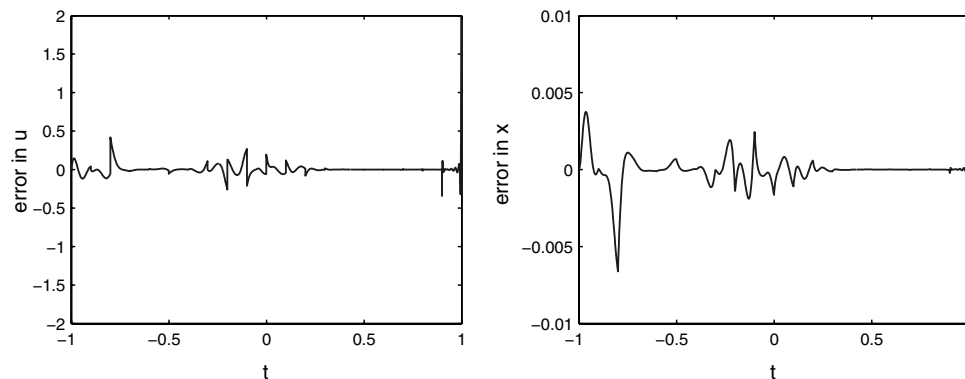Fig. 8   Optimal solution after antialiasing (dashed line) overlaid with the exact solution (solid line).

Fig. 9   Errors in the a^2B solution to problem T.

Euler methods are frequently used in testing large sparse NLPs, in electrical engineering, and the so-called model-predictive control methods. Because convergence of the Euler method is only $O(h)$, it is apparent that it is not a good method to be used in conjunction with our a^2B algorithm. The Hermite–Simpson direct method, initiated by Hargraves and Paris [38] based on the work of Dickmanns and Well [39], is widely used and implemented in the software packages OTIS [20] and SOCS [21]. Although the Hermite–Simpson method is a three-stage RK method, it turns out that it is $O(h^4)$ in the Hager family of Runge–Kutta methods. That is, one can readily verify that the RK coefficients of the Hermite–Simpson method satisfy the standard $O(h^4)$ conditions of Butcher [40] in addition to those of Hager [7],

$$\sum d_i^2/b_i = 1/3, \quad \sum d_i^3/b_i^2 = 1/4, \quad \sum b_i c_i a_{ij} d_j/b_j = 5/24$$

$$\sum c_i d_i^2/b_i = 1/12, \quad \sum d_i a_{ij} d_j/b_j = 1/8$$

where

$$c_i := \sum_{j=1}^{3} a_{ij} = (0, 1/2, 1) \qquad d_j := \sum_{i=1}^{3} b_i a_{ij} = (1/6, 1/3, 0)$$

Hence, under his assumptions, Hager's theorem guarantees a convergence of the Hermite–Simpson method to the optimal solution at an $O(h^4)$ rate [7]. A theorem similar to Hager's has been obtained by Gong et al [19,41,42] for the Legendre pseudospectral method. It is important to note that the technical assumptions necessary for the validity of Hager's theorem are quite different from those required for the convergence of the Legendre pseudospectral method. If a given problem meets all the conditions for both theorems to hold, then the Legendre pseudospectral method provides superior convergence properties with respect to any RK method. Numerical experiments confirm this point [42–45]; consequently, the Legendre pseudospectral method has become an attractive choice for an upgrade of the OTIS software package [12]. Of course, the software package DIDO uses pseudospectral methods exclusively, and independent tests [30,46,47] of DIDO with other tools have consistently shown its superior performance. Despite these favorable results for pseudospectral methods, we note that it is not a panacea. Thus, it is quite possible for a given problem to satisfy the assumptions of Hager's theorem but fail those of the Legendre pseudospectral method. In such situations, the Hermite–Simpson method can be superior to the Legendre pseudospectral method.

The strongest convergence theorems provide rates of convergence of estimated errors and not the values of the true errors. That is, if the rates of convergence of any two methods are the same, they are mathematically equivalent even if anecdotal examples show differences in the values of true errors. Thus, a convergence theorem serves the purpose of an a priori selection criterion for the a^2B algorithm. Because no convergence theorem provides the true error values, it nonetheless leads to the following set of self-tests on the accuracy of results.

A. Self-Test on Optimality

For the a^2B algorithm to work as a pure antialiasing scheme, we require three conditions as follows:

1) the discrete solution pairs, $\{\mathbf{x}_i, \mathbf{u}_i\}_{i=0}^{n}$, must exactly lie on the optimal trajectory;

2) the continuous-time reconstructions of the controls, $\mathbf{u}^i(t)$, $t \in [t^{i-1}, t^i]$, must be exact over each Bellman segment; and

3) the propagation of the initial value problem (see step 3) must be exact.

In practice, all three conditions will be violated to varying levels of precision. The combined effect of all these violations from a desired precision can be detected, and thereafter corrected, using the following tests.

If $\{\mathbf{x}_i, \mathbf{u}_i\}_{i=0}^{n}$ from step 1 does indeed lie on the exact optimal trajectory, and if there were no errors whatsoever in generating the

Bellman-sequenced trajectory, $\mathbf{x}_B(t)$, $t \in [t_0, t_f]$ (see step 5), then, according to Bellman's principle of optimality we must have

$$\mathbf{x}_B(t_i) = \mathbf{x}_i, \qquad i = 0 \cdots n \tag{2}$$

Any violation of Eq. (2) is, therefore, a violation on optimality. A measure of this violation can be written in terms of any error norm; for example,

$$\Delta x_{\text{opt}}^n := \frac{1}{n} \sum_{i=0}^{n} \|\mathbf{x}_B(t_i) - \mathbf{x}_i\| \tag{3}$$

When convergence is assured by a proper method, we have $\lim_{n \to \infty} \Delta x_{\text{opt}}^n \to 0$. In practice, $\Delta x_{\text{opt}}^n = 0$ cannot be achieved with infinite precision; hence, we must limit n to some sufficiently large value that leads to a Δx_{opt}^n less than some desired tolerance. Because every element on the right-hand side of Eq. (3) is already computed as part of the a^2B algorithm, it is clear that this equation provides a self-test on optimality in terms of first principles itself whenever $\{\mathbf{x}_i, \mathbf{u}_i\}_{i=0}^{n}$ is recalculated.

In the same spirit, an estimate of suboptimality is obtained from the differences in the values of the Bellman-sequenced cost functional and the discrete cost of each run. Denoting these quantities as $\mathcal{J}_B$ and $\mathcal{J}^n$, we have

$$\Delta \mathcal{J}_{\text{opt}}^n := |\mathcal{J}_B - \mathcal{J}^n| \tag{4}$$

as a measure of suboptimality. As with Eq. (3), we have $\lim_{n \to \infty} \Delta \mathcal{J}_{\text{opt}}^n \to 0$.

B. Criterion for Selecting Bellman Segments

Suppose that the initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$ is given and/or computed exactly (i.e., no round off errors in the ideal case). Then, from step 3 of the a^2B algorithm, we have

$$\mathbf{x}^1(t^1) = \mathbf{x}_0 + \int_{t_0}^{t^1} \mathbf{f}(\mathbf{x}^1(t), \mathbf{u}^1(t)) \, dt \tag{5}$$

where $t \mapsto \mathbf{u}^1$ is the control interpolated over $[t_0, t^1]$ using $\{\mathbf{u}_i\}_{i=0}^{n}$ as data points. Note that this interpolation need not be continuous, in which case, Eq. (5) defines an absolutely continuous (Carathéodory) solution [48] for Eq. (1). Allowing discontinuities in $\mathbf{u}^1(\cdot)$ is done not merely for greater mathematical generality, but to make our algorithm applicable to many practical problems. A case in point is a typical orbit transfer problem that has discontinuous optimal controls in the form of switches (see Fig. 2). Imposing continuity conditions for the controls will render such solutions nonoptimal. Regardless, $\mathbf{x}^1(\cdot)$ will not be perfectly optimal since integration in Eq. (5) is not perfect even if $t \mapsto \mathbf{u}^1$ is exactly optimal (which can happen in practice, as in bang-bang control systems). Thus, from Bellman's principle, we have

$$\mathbf{x}^1(t^1) \neq \mathbf{x}^*(t^1) \tag{6}$$

where $\mathbf{x}^*(t^1)$ is the theoretically exact (and frequently, unknown) optimal state at time t^1. Nonetheless, we can control the error $\|\mathbf{x}^1(t^1) - \mathbf{x}^*(t^1)\|$ to within any given $\epsilon > 0$ if the interpolation error in the controls $\|\mathbf{u}^1(\cdot) - \mathbf{u}^*(\cdot)\|_{L^\infty}$ and the Bellman segment $t^1 - t_0$ are sufficiently small. A precise statement of this intuition is given by the following theorem.

Theorem 2: Let $\mathbf{f}(\mathbf{x}, \mathbf{u})$ be Lipschitz continuous in the domain of feasible $(\mathbf{x}, \mathbf{u})$ with Lipschitz constants $Lipf_x$ and $Lipf_u$, respectively. Let

$$\delta := \|\mathbf{u}^1(\cdot) - \mathbf{u}^*(\cdot)\|_{L^\infty} \left(\frac{Lipf_u}{Lipf_x} \right) \tag{7}$$

For any given $\epsilon > 0$, if

$$(t^1 - t_0) \leq \frac{W(r)}{Lipf_x} \tag{8}$$

then $\|\mathbf{x}^1(t^1) - \mathbf{x}^*(t^1)\| \le \epsilon$, where $W(r)$ is the Lambert W function (see the Appendix), and r is the ratio, ϵ/δ.

Proof: See the Appendix.

Remark 2.1: If $\epsilon = \delta$, we have $W(1) = 0.56714, \ldots$. Hence, under this special situation, if

$$(t^1 - t_0) \le \frac{1}{2Lipf_x} \Rightarrow (t^1 - t_0) < \frac{W(1)}{Lipf_x}$$

Thus, it appears that, analogous to the Nyquist–Shannon guidelines in choosing a sampling frequency, $Lipf_x$ acts as a fundamental frequency of the dynamical system that, in conjunction with Eq. (8), provides a measure for choosing Bellman segmentations.

C. Accuracy of Controls

As noted in Sec. IV.B, a high accuracy in reconstructing the states is possible by allowing the controls to be interpolated discontinuously over each Bellman segment, $[t^i, t^{i+1}]$. Nonetheless, as a result of Eq. (6), we have

$$\mathbf{u}^2(t^1) \ne \mathbf{u}^1(t^1) \qquad (9)$$

in general. Although it is possible to solve optimization problems by enforcing the controls to be continuous across the Bellman segments, this may reduce accuracy for the same reason that the optimal control may, in fact, be discontinuous at t^1. It turns out that allowing the controls to be discontinuous across the Bellman segments has a very practical advantage in terms of a yet another self-test on optimality. That is, suppose that the (unknown) optimal control was continuous at t^1. Then, Eq. (9) is an indicator of nonoptimality, and the quantity $\|\mathbf{u}^2(t^1) - \mathbf{u}^1(t^1)\|$ can be taken as a measure of the nonoptimality of the control. Thus, in addition to Eqs. (3) and (4), we take

$$\Delta u_{\text{opt}}^n := \sum_{k=1}^{N_B-1} \|\mathbf{u}^{k+1}(t^k) - \mathbf{u}^k(t^k)\| \qquad (10)$$

as a third indicator and measure of suboptimality of the overall Bellman-sequenced trajectory.

It is very important to note that we do not take Eq. (10) as a sole test of optimality but as part of the menu of tests resulting from a first-principles' application of Bellman's principle. Thus, if the optimal controls are discontinuous at exactly the endpoints of the Bellman segments, Eq. (10) will provide an incorrect measure of suboptimality. A simple remedy to solve this problem would be to implement the a^2B algorithm using a different set of Bellman segments so that in the unlikely event that the optimal controls were discontinuous at exactly the segment endpoints of the first implementation, they would be continuous at the endpoints of the second segmentation. This assertion can be guaranteed if the optimal controls are discontinuous over a finite number of points and do not exhibit the Fuller phenomenon [49]. Hence, we include this condition as part of the assumption for the validity of the a^2B algorithm.

D. Universality of the Optimality Tests

The three optimality tests given by Eqs. (3), (4), and (10) can be used to test the optimality of any computed trajectory whether or not antialiasing is sought. Thus, unlike tests on the Pontryagin extremality that require computation of the dual variables, the tests we propose require only primal variables. Consequently, if a direct method is used, and a covector mapping theorem is unavailable to determine the duals, the optimality of the computed trajectory can still be quickly checked by the primal tests we proposed in this paper. Clearly, if dual variables are also available, then a robust verification of optimality of the computed solutions is possible by combining the battery of tests provided by the minimum principle in concert with Eqs. (3), (4), and (10). Furthermore, unlike the minimum principle which provides a negative test on the optimality of the solution, Eqs. (3), (4), and (10) provide a quantitative measure of deviation from optimality. These ideas are further illustrated in Sec. VI.

E. Enhancing the a^2B Algorithm via Node Clustering

From the developments of the preceding sections, it is clear that the proposed ideas can be used in conjunction with any computational method provided that it meets the essential qualifications of pointwise convergence. For example, the a^2B algorithm can be implemented in conjunction with the Hermite–Simpson method. Because a motivation for the a^2B algorithm is a long time span, it is clear that we would benefit from long Bellman segments by lessening the effects of accumulation errors. Furthermore, according to Theorem 2, the error per Bellman segment depends upon two critical quantities, namely, 1) the problem dynamics (through its Lipschitz constants), and 2) the accuracy of the control interpolation [cf. Eq. (7)].

Thus, for any given problem (i.e., given Lipschitz constants), the accuracy of the optimal state trajectory, as measured by ϵ, depends upon the accuracy of the control interpolation, as measured by δ. To have a small δ, we need to capture the high-frequency components of the control trajectory. According to Theorem 1, this normally means that we need to have a high sampling rate (i.e., fine mesh) over the entire time span of control interpolations. However, our proposed a^2B algorithm only uses the portion of the control trajectory $\{\mathbf{u}_i\}_{i=0}^n$ that falls within the time interval where the differential equation is propagated, say, $[t_0, t^1]$ (see step 3 of the a^2B algorithm). This implies that under the continuing assumption of pointwise convergence, it would be desirable to have a mesh refinement strategy that is biased to produce a finer grid close to the initial time. In examining Figs. 4, 6, and 7, it is clear that this condition is met by the method we have used to solve problem T. This is the Legendre pseudospectral method. In fact, pseudospectral methods based on polynomial interpolation have a natural property of node clustering at the initial time [31]. That is, the average spacing between points over $[t_0, t^1]$ is $O(n^{-2})$ in contrast to a uniform grid which produces a mesh size of $O(n^{-1})$. This property of pseudospectral methods is the reason why all three interpolations of Fig. 4 generate negligible errors over the initial time period of approximately $[-1, -0.8]$ (see Fig. 5) as all the high-frequency components of the control signal are sufficiently represented by the natural node clustering of the Legendre pseudospectral method. This last statement is evident by an inspection of Fig. 6.

V. Low-Thrust Preliminaries

Low-thrust technology is substantially varied. The design and capabilities of the engine, the physics of the propulsion, and the type of power plant all affect the optimal trajectory design. For detailed mission analysis, these complexities must all be considered [50–53]. During preliminary stages of mission design, it is desirable to perform low-thrust trajectory analysis without tying it to a particular propulsion system so that the merits of the trajectory can be independently assessed. For minimum-fuel considerations, this translates to formulating space trajectory optimization problems as L^1-optimal control problems [8]. A p–q family of coplanar L^1-optimal control problems can be cast as (see Fig. 10)

$$\mathbf{x}^T := [r, \theta, v_r, v_t] \qquad \mathbf{u}^T := [u_r, u_t] \qquad \mathbf{u} \in \mathbb{U}(q)$$

$$\mathbb{U}(q) := \{\mathbf{u} \in \mathbb{R}^2 : \|\mathbf{u}\|_q \le u_{\max}\}$$

$$O(p) \begin{cases} \text{Minimize} & J_p[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = \int_{t_0}^{t_f} \|\mathbf{u}(t)\|_p \, dt \\ \text{Subject to} & \dot{r} = v_r \\ & \dot{\theta} = \frac{v_t}{r} \\ & \dot{v}_r = \frac{v_t^2}{r} - \frac{\mu}{r^2} + u_r \\ & \dot{v}_t = -\frac{v_r v_t}{r} + u_t \\ & \mathbf{e}_0(t_0, \mathbf{x}_0) = \mathbf{0} \\ & \mathbf{e}_f(\mathbf{x}_f) = \mathbf{0} \\ & t_f \le t^U \end{cases}$$

where $t^U \ge t_{\min}$ is an upper bound on the final time,
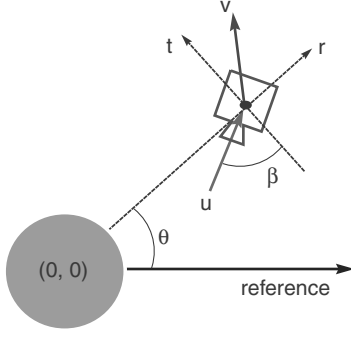
Fig. 10 Schematic of the parameters for problem $O(2)$.

$$\frac{1}{p} + \frac{1}{q} = 1, \qquad p \in \{1, 2\}$$

and the functions for the endpoint conditions,

$$\mathbf{e}_0(t, \mathbf{x}) := \begin{pmatrix} t \\ a_0[(v_r^2 + v_t^2)r - 2] + r \\ r[1 + e_0 \cos(\theta - \omega_0)] - (v_t r)^2 \\ v_r[1 + e_0 \cos(\theta - \omega_0)] - e_0 v_t \sin(\theta - \omega_0) \end{pmatrix}$$

$$\mathbf{e}_f(\mathbf{x}) := \begin{pmatrix} a_f[(v_r^2 + v_t^2)r - 2] + r \\ r[1 + e_f \cos(\theta - \omega_f)] - (v_t r)^2 \\ v_r[1 + e_f \cos(\theta - \omega_f)] - e_f v_t \sin(\theta - \omega_f) \end{pmatrix}$$

describe the initial and final manifolds for problem $O(p)$ in terms of the initial (a_0, e_0, ω_0) and final (a_f, e_f, ω_f) orbital elements, respectively. Except for its resemblance to the dynamic model, this problem formulation is different in every respect when compared to the continuous-thrust problem posed by Moyer and Pinkham [2] and popularized by Bryson and Ho [3]. Surprisingly, what is not widely discussed in many texts (barring optimization books [54]) is the potency of scaling and balancing equations. As a matter of fact, many of the reported difficulties in low-thrust optimization can be immediately addressed by a proper scaling of the optimal control problem [35,55]. Hence, we address this issue first.

A. Scaling and Balancing Equations

Canonical units are frequently used to scale the orbit dynamical equations [2–4]. Although useful in some problem analyses, these units may not be entirely appropriate for certain applications. In such situations, it is useful to solve problems using "designer units." That is, we would like to scale each of the N_x variables in $\mathbf{x} \in \mathbb{R}^{N_x}$, the N_u variables in $\mathbf{u} \in \mathbb{R}^{N_u}$, and time t in some arbitrary, but appropriately chosen units. Thus, we define

$$\bar{r} := r/R, \qquad \bar{\theta} := \theta/A, \qquad \bar{v}_r := v_r/V_r, \qquad \bar{v}_t := v_t/V_t$$

$$\bar{u}_r := u_r/U_r, \qquad \bar{u}_t := u_t/U_t, \qquad \bar{t} := t/T$$

where R, A, V_r, V_t, U_r, U_t, and T are arbitrary numbers chosen to serve a desirable outcome. These numbers are "units" for the indicated quantities. We now rewrite problem $O(p)$ in terms of $\bar{r}$, $\bar{\theta}$, $\bar{v}_r$, etc. This is done easily by the inverse transformation,

$$r = \bar{r}R, \qquad \theta = \bar{\theta}A, \qquad v_r = \bar{v}_r V_r, \qquad v_t = \bar{v}_t V_t$$

$$u_r = \bar{u}_r U_r, \qquad u_t = \bar{u}_t U_t, \qquad t = \bar{t}T$$

It is trivial to show that $\dot{r} = v_r$ is now replaced by

$$\frac{d\bar{r}}{d\bar{t}} = \left(\frac{V_r T}{R}\right)\bar{v}_r \qquad (11)$$

In the same spirit, the equation for $\dot{v}_t$ is replaced by

$$\frac{d\bar{v}_t}{d\bar{t}} = -\left(\frac{V_r T}{R}\right)\frac{\bar{v}_r \bar{v}_t}{\bar{r}} + \left(\frac{U_t T}{V_t}\right)\bar{u}_t \qquad (12)$$

Thus, the dynamical equations are modified as a result of introducing scaling factors R, T, etc. These designer units (or scaling factors) can now be chosen in a manner that balances the variables and/or the equations conducive to numerical analysis. Note that there is no need to choose the velocity units, V_r and V_t, to be the same, nor do they need to be equal to distance units divided by time. Such a practice leads to canonical units:

$$V_r = V_t = \frac{R}{T} = \frac{\mu T}{R^2} = TU_t = TU_r$$

That is, the units for velocity, acceleration, etc., are all derived from the units of the fundamental quantities such as distance and time, and the corresponding units are chosen to render the scaled equations identical to the unscaled equations. Although such scaling based on units of the fundamental quantities has highly desirable properties and is physically appealing, it is not necessarily the best choice for computation, particularly for low-thrust applications. For example, in terms of canonical units based on the Earth's radius of $R = 6378$ Km, a unit of acceleration is the standard $g_0 = 9.8$ m/s^2. Thus a force unit for a 1000 kg spacecraft would be 9800 N. Now, an NSTAR engine [56] generates about 90 mN of maximum thrust force. Thus, in canonically scaled units of force, the NSTAR engines generate 9×10^{-6} units of maximum force which is, clearly, imbalanced with the other quantities (e.g., $\bar{r} \sim 1$). Now, consider a designer unit where the unit of force is set to 90 mN. In this case, the maximum NSTAR engine thrust is just one force unit. Of course, the dynamical equations are no longer in the standard form as suggested in problem $O(p)$; rather, they have to be rewritten in the balanced form as in Eqs. (11) and (12).

The designer units used in solving all the examples in the remaining part of the paper are

$$R = r_0 \quad A = 10 \quad V_r = 0.01\sqrt{\mu/r_0} \quad V_t = 0.1\sqrt{\mu/r_0}$$

$$U_r = 0.001(\mu/r_0^2) \quad U_t = 0.001(\mu/r_0^2) \quad T = \sqrt{r_0^3/\mu} \quad (13)$$

where r_0 is the initial value of $r(t)$, whatever it might be in traditional units. These noncanonical units were chosen simply by inspection of the problem data.

B. Time-Bound Time-Free Trajectories

From the physics of the problem, it is apparent that a time-free minimum-fuel trajectory could require an infinite time. An infinite horizon problem not only creates a computational problem, it also generates some deep theoretical issues on existence and uniqueness of solutions. These problems are simply circumvented by imposing an upper bound on the transit time by setting $t_f \leq t^U$ (with $t_0 = 0$) as evident in the problem formulation of problem $O(p)$. Obviously, $t^U \geq t_{\min}$, where $t_{\min}$ is the cost obtained by solving the corresponding minimum-time problem. This is one reason why we discuss minimum-time trajectories in the next section. In any event, it is apparent that if $t^U = t_{\min}$, then the minimum-time and minimum-fuel trajectories are the same. When problem $O(p)$ is parameterized by t^U [in which case, we would write it as problem $O(p; t^U)$], the resulting solutions are not "linear" for small perturbations of $t^U \in [t_{\min}, \infty)$. That is, as t^U is varied continuously upward from $t_{\min}$, the solutions exhibit "jumps" as t^U crosses certain thresholds. This is essentially a result of the fact that the problem is nonconvex (any nonlinear equality implies nonconvexity). This property of solution jumps for nonconvex problems even when nonbinding constraints are removed or introduced is a well-known phenomenon in optimization [54]. Thus, for example, if the optimal final time t_f^* in a time-bound minimum-fuel trajectory is such that $t_f^* < t^U$ (as is to be generally expected), then, despite the fact that the condition $t_f \leq t^U$ is nonbinding, increasing t^U (or removing it) does not produce the same solution (both theoretically and computationally). Thus, a proper

way to characterize these trajectories is not in the classical sense of time-free trajectories but as time-bound time-free trajectories.

VI.   Illustrative Results

Having established the importance of minimum-time trajectories as a first step toward solving minimum-fuel problems (especially for industrial strength problems), we address this problem first.

A.   Minimum-Time Trajectories

For the purpose of illustrating the main ideas, we choose the initial and final orbits to be circular and take the control space $\mathbb{U}$ to be a Euclidean box (this is the same as setting $q = \infty$ in $\mathbb{U}(q)$). We use the familiar canonical units for a physical description but use Eq. (13) to balance the problem. Thus, we have

$$\mathbf{x}^T := [r, \theta, v_r, v_t] \qquad \mathbf{u}^T := [u_r, u_t]$$

$$\mathbf{u} \in \mathbb{U}(\infty) := \{\mathbf{u} \in \mathbb{R}^2 : \|\mathbf{u}\|_\infty \le 0.01\}$$

$$(\min T) \begin{cases} \text{Minimize} & J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = t_f \\ \text{Subject to} & \dot{r} = v_r \\ & \dot{\theta} = \frac{v_t}{r} \\ & \dot{v}_r = \frac{v_t^2}{r} - \frac{1}{r^2} + u_r \\ & \dot{v}_t = -\frac{v_r v_t}{r} + u_t \\ & (r, \theta, v_r, v_t)(0) = (1, 0, 0, 1) \\ & (r, v_r, v_t)(t_f) = (4, 0, 0.5) \end{cases}$$

A candidate discrete-time optimal transfer trajectory for $n = 30$ nodes is shown in Fig. 11. It is clear that this is not even a physically meaningful trajectory. We now apply the a^2B algorithm using this purportedly poor solution. The resulting antialiased trajectory is shown in Fig. 12. It is apparent that this trajectory is more realistic. The closed-loop control corresponding to this antialiased solution is shown in Fig. 13. Not only is this antialiased solution meaningful, it is in fact very close to an optimal solution. To illustrate this point in terms of the theory developed in Sec. IV, consider applying the optimality test given by Eq. (3) to explore the optimality of the 30-node solution. This generates

$$\Delta x_{\text{opt}}^{30} = \frac{1}{30} \sum_{i=0}^{30} \|\mathbf{x}_B(t_i) - \mathbf{x}_i\|_2 = 0.0317$$

The small number [in comparison to $\|\mathbf{x}_B(\cdot)\|_{L^2}$] indicates that the 30-node solution may be very close to the optimal solution. Using the computed costs of $\mathcal{J}^{30} = 48.142$ and $\mathcal{J}_B = 48.346$, we have a measure of suboptimality given by Eq. (4) as

$$\Delta \mathcal{J}_{\text{opt}}^{30} := 0.204 \; TUs \tag{14}$$

As theorized in Sec. IV, this number would not have been small (in comparison to either $\mathcal{J}^{30}$ or $\mathcal{J}_B$), if the antialiased solution was not
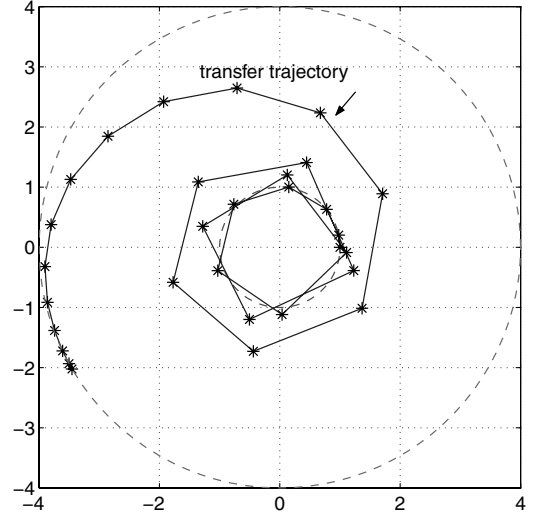

Fig. 11   A 30-node open-loop transfer trajectory for problem min T.
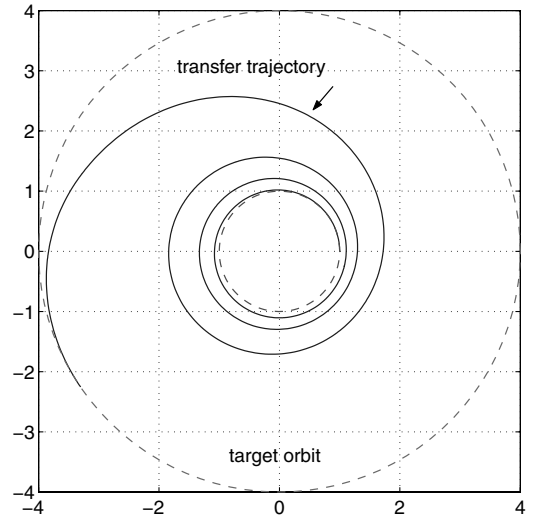

Fig. 12   An antialiased 30-node closed-loop transfer trajectory for problem min T.

very close to the optimal. Simply put, Figs. 12 and 13 illustrate Bellman's principle in action for a converged solution. To further validate convergence, we apply the a^2B algorithm with 50 nodes. The error $\Delta x_{\text{opt}}^{50}$ reduces to 0.0243 and the new cost is $\mathcal{J}^{50} = 47.809 < \mathcal{J}^{30}$. Both these quantities indicate convergence. *Note that all of these tests are only primal in nature and do not require dual variables.*
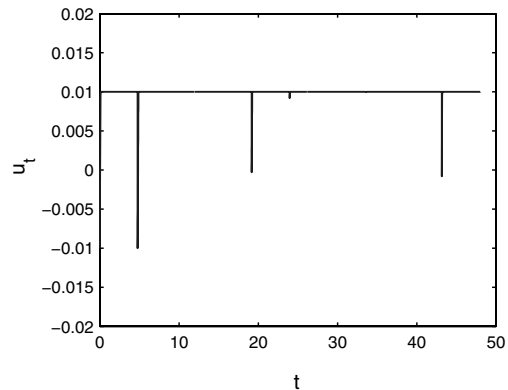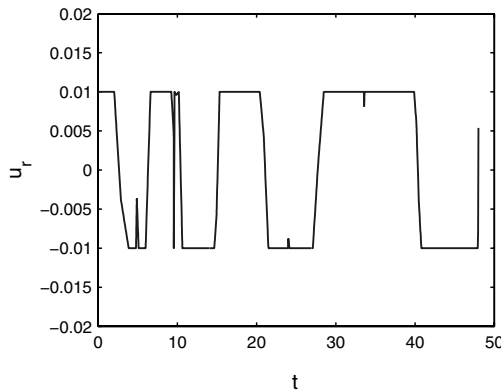

Fig. 13   An antialiased 30-node closed-loop control solution for problem min T.
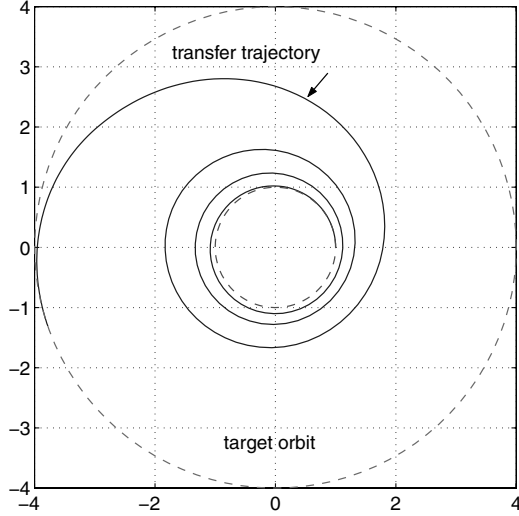
Fig. 14   A 300-node open-loop transfer trajectory for problem min T.

$$H(\lambda, \mathbf{x}, \mathbf{u}) := \lambda_r v_r + \lambda_\theta \frac{v_t}{r} + \lambda_{v_r}\left(\frac{v_t^2}{r} - \frac{1}{r^2} + u_r\right)$$

$$+ \lambda_{v_t}\left(-\frac{v_r v_t}{r} + u_t\right)$$

Minimizing H over $\mathbb{U}(\infty)$, we get

$$u_r(t) = \begin{cases} -0.01 & \text{if } \lambda_{v_r}(t) \geq 0 \\ \text{singular} & \text{if } \lambda_{v_r}(t) \equiv 0 \\ +0.01 & \text{if } \lambda_{v_r}(t) \leq 0 \end{cases} \tag{15}$$

$$u_t(t) = \begin{cases} -0.01 & \text{if } \lambda_{v_t}(t) \geq 0 \\ \text{singular} & \text{if } \lambda_{v_t}(t) \equiv 0 \\ +0.01 & \text{if } \lambda_{v_t}(t) \leq 0 \end{cases} \tag{16}$$

The switching functions together with the candidate optimal controls are shown in Figs. 16 and 17. It is clear from these plots that the controls indeed satisfy the Hamiltonian minimization conditions of Eqs. (15) and (16).

The computed minimum transfer time for the 300-node solution is 47.706 TUs while the corresponding transfer time for the antialiased 30-node solution is 48.346 TUs. That is, the cost of the antialiased solution is larger than the "full scale" solution by only about 1.34%. That the extremal satisfies Bellman's principle within numerical tolerances indicates that we have indeed found an optimal solution with only 30 nodes. All of these points clearly validate the superior performance of the a^2B algorithm for orbit transfer applications.

We now set the control bound u_{max} to 0.002 and resolve problem min T. It is apparent from the physics of the problem that the new trajectory will take many more revolutions than when u_{max} was set to 0.01. This is also borne out by the solution we obtained from our a^2B algorithm as shown in Fig. 18. This solution was obtained using 100 nodes. By simple analogy to the higher-thrust solution, it can be argued that the 100-node antialiased solution is equivalent to a 1000-node generic solution. The controls corresponding to Fig. 18 are shown in Fig. 19. The average run time to obtain a 100-node solution (also with unoptimized code running on low-end machines) was about 6 s. As a matter of completeness a plot of the state trajectories is shown in Fig. 20. This plot features many of the information-theoretic concepts associated with high-frequency content and potential aliasing problems discussed earlier.

For the problem under study, the average run time for generating open-loop time-optimal solutions on a low-end Pentium M (1.6 GHz with 256 MB of RAM) laptop was about 0.3 s. That is, even without optimizing the code itself, solutions can be obtained in fractions of a second. As it is not too difficult to show that $Lipf_x$ for problem min T is substantially less than $\frac{1}{2(0.3)}$ s^{-1}, the 0.3 s of run time easily meet the condition of Theorem 2 for declaring the antialiased solution shown in Figs. 12 and 13 as a closed-loop solution. Further details on this new way of generating closed-loop solutions are described in [24,25].

The apparent infeasibility of the trajectory shown in Fig. 11 is, in fact, due to an aliasing problem. To illustrate this point, a 300-node open-loop state trajectory and control solution are shown in Figs. 14 and 15. Comparing Figs. 12–14 and Figs. 13–15, it is clear that the controls and trajectories are nearly the same. In the case of the 300-node solution, the number of optimization variables is $N_n(N_x + N_u) + 1 = 300(4 + 2) + 1 = 1801$, while the corresponding number for the 30-node solution is 181, obviously, a reduction by an order of magnitude. Although 1801 optimization variables is not large by current standards, the purpose of this example is similar to that of the test problem, problem T, discussed earlier, that is, to show that we can obtain nearly the same solution with a substantially lower number of optimization variables. This is one of the main reasons why multi-agent problems [8,27–30] can be solved with the same ease as single-agent problems.

The two solutions shown in Figs. 12–14 would not have been nearly the same in case they were not nearly optimal. As a matter of further validation and comparison, consider applying standard dual optimality tests generated by an application of Pontryagin's minimum principle. The control Hamiltonian for problem min T is given by

B.   Minimum-Fuel Trajectories

We now solve problem $O(p)$ for $p = 1$ and 2 with the rest of the parameters set the same as problem min T. To bound the scope of the paper, we limit our discussions to the case when $u_{max} = 0.01$ for both $\mathbb{U}(\infty)$ and $\mathbb{U}(2)$.

Consider problem $O(1)$. This problem represents a minimum-fuel orbit transfer problem for a spacecraft equipped with multiple thrusters [8]. We choose
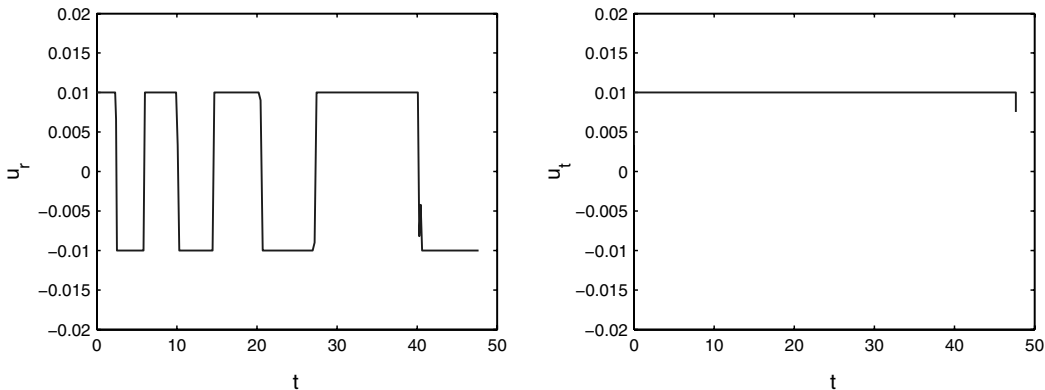

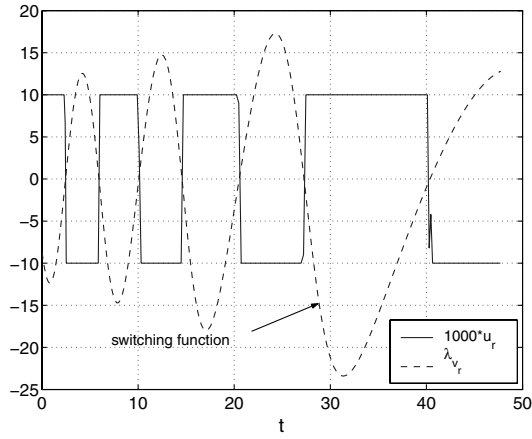Fig. 15   A 300-node open-loop control solution for problem min T.

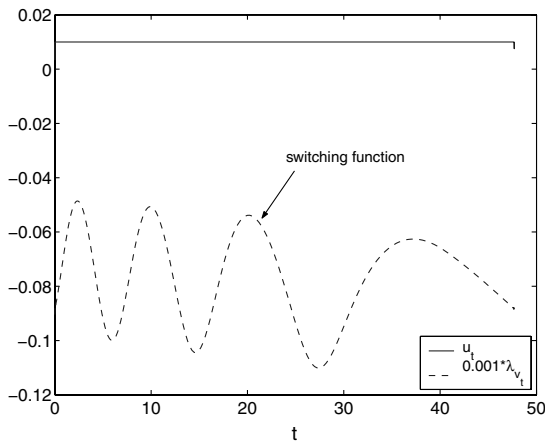Fig. 16   Switching function and $u_r(t)$ for problem $\min T$.

Fig. 19   Low-thrust antialiased 100-node controls corresponding to Fig. 18.

Fig. 17   Switching function and $u_t(t)$ for problem $\min T$.

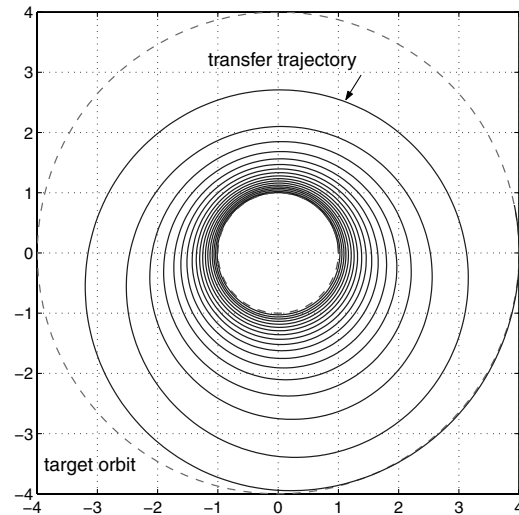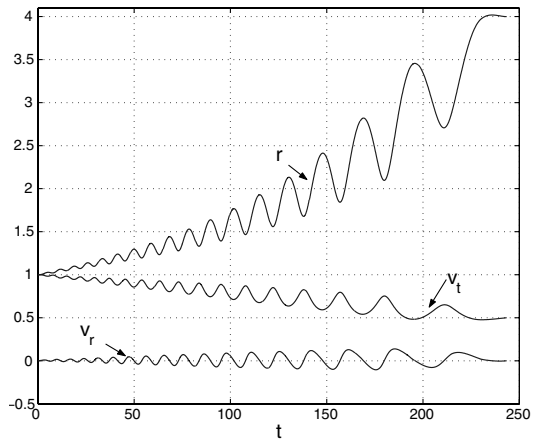Fig. 20   Antialiased 100-node state trajectories corresponding to Fig. 18.

Fig. 18   Low-thrust antialiased transfer trajectory for problem $\min T$ modified for $u_{\max} = 0.002$.

$$t^U = 55 > 48.142 = \mathcal{J}^{30}_{\min T}$$

to ensure that any potential infeasibility is not due to an improper selection of an upper bound on the maximum allowable transfer time. We now use the subscript $\min T$ on the value of the cost function for problem $\min T$ to distinguish it from the value of the cost function for problem $O(p)$.
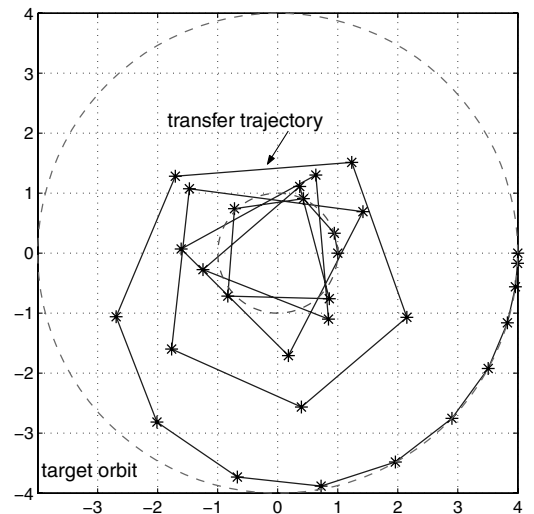

Fig. 21   A 30-node open-loop transfer trajectory for problem $O(1)$.

Shown in Fig. 21 is the computed optimal trajectory for $n = 30$ without the benefit of the a^2B algorithm. As in the minimum-time case discussed in Sec. VI.A, it is clear that this is not even a physically meaningful trajectory. On the other hand, when we apply our a^2B algorithm with the same $n = 30$, the result is dramatically different as
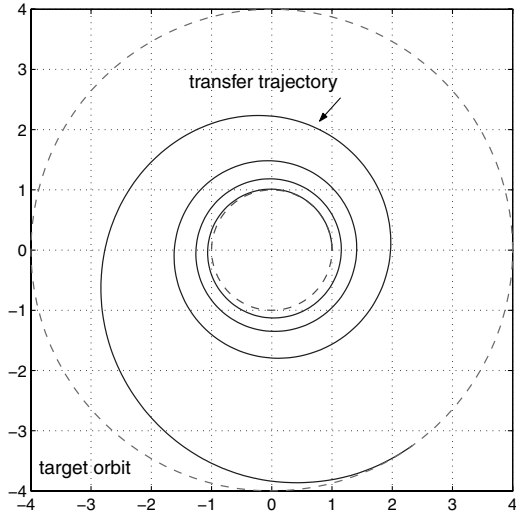
Fig. 22 An antialiased 30-node closed-loop transfer trajectory for problem $O(1)$.

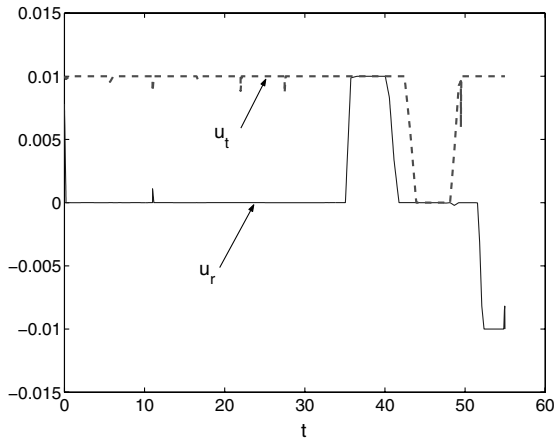Fig. 24 Antialiased 30-node closed-loop transfer trajectory for problem $O(2)$.

Fig. 23 An antialiased 30-node closed-loop control solution for problem $O(1)$.

evident in Figs. 22 and 23. Because the bound t^U is active (see Fig. 23), this solution corresponds to a time-fixed minimum-fuel orbit transfer. In other words, the solution shown in Figs. 22 and 23 corresponds to a minimum-fuel solution incurred at a penalty of

$$55.000 - 48.142 = 6.858 \, TUs$$

or about 14% increase in transfer time. The fuel cost for problem $O(1)$ was computed to be

$$\mathcal{J}_1 = 0.57815$$

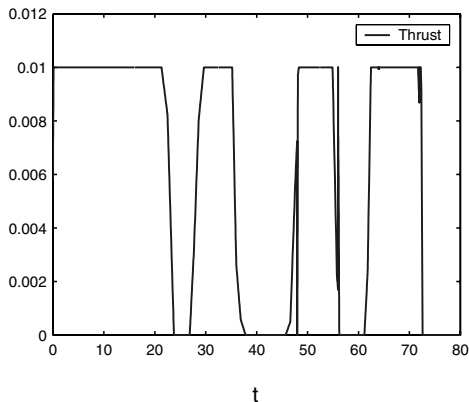whereas the corresponding fuel cost for problem $\min T$ was determined to be equal to 0.95050. Consequently, we conclude that a fuel savings of about 39% is achieved at the price of about 14% increase in transfer time.

Finally, we consider problem $O(2)$ and choose $t^U = 100$. This problem represents a minimum-fuel orbit transfer problem for a spacecraft equipped with a single thruster (see Fig. 10). Note also that the cost function in problem $O(2)$ is not quadratic; in fact, quadratic costs do not minimize fuel [8]. As in the previous cases, the computed optimal trajectory for $n = 30$ appears similar to Fig. 21; however, after an application of our a^2B algorithm and keeping $n = 30$, the result is physically realizable as evident in Fig. 24. The corresponding controls are shown in Fig. 25. In conformance with Fig. 10, the controls are plotted in the usual magnitude and direction variables.

It is apparent from Fig. 25 that the bound $t^U = 100$ is not active. The computed optimal "free" time was $t_f^* = 80 < t^U$ with fuel cost $\mathcal{J}_2 = 0.49327$. In terms of the notion developed in Sec. V.B, this is a time-bound time-free minimum-fuel solution and not a classical time-free solution. A fairly comprehensive discussion of these new principles on minimum-fuel optimal control and their relationships to the Bellman and Pontryagin frameworks may be found in [8]. Further results on problems $\min T$, $O(1)$, and $O(2)$ are discussed by Mendy [29] who also uses equinoctial element sets to capture the full scope of three-dimensional orbital maneuvers for multiple spacecraft.

VII. Conclusions

The antialiasing Bellman (a^2B) algorithm proposed in this paper solves low-thrust trajectory optimizations with embarrassing ease.
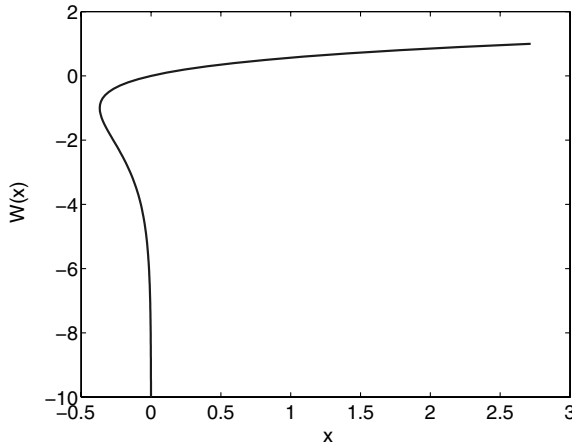

Fig. 25 An antialiased 30-node closed-loop control solution for problem $O(2)$.
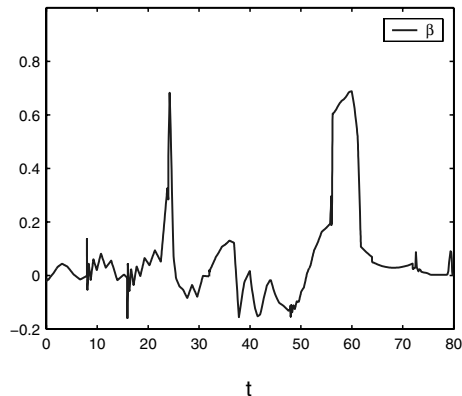
Fig. 26 Plot of the Lambert *W* function.

Although all the solutions were obtained using pseudospectral methods, the a^2B algorithm is, in principle, *independent* of pseudospectral methods. Thus, the algorithm can be implemented in conjunction with any existing tools, methods, or software that are capable of legitimately solving low-scale trajectory optimization problems. Because these problems can typically be solved in fractions of a second, the a^2B solutions are in fact closed-loop solutions for most applications. It is this closed-loop feature of the algorithm that provides a high level of accuracy via an antialiasing process.

The essential principle behind the simplicity of the proposed method can be framed in terms of fundamental optimal control theory as follows: in the Hamilton–Jacobi–Bellman theory, solutions are sought over the entire domain (i.e., the entire state space) whereas in the Pontryagin framework, solutions are sought only over a line or a tube. In the proposed framework, solutions are sought merely near the initial conditions whereas the full solution is sequentially determined by moving the initial conditions toward the final conditions by a first-principles' application of Bellman's principle of optimality. Consequently, self-tests on optimality follow directly from first principles without the need for computing costates or solving the Hamilton–Jacobi–Bellman equation.

Appendix

I. Lambert *W* function

The multivalued function $x \mapsto W(x)$, given implicitly by

$$x = W(x)e^{W(x)} \tag{A1}$$

is called the Lambert *W* function. A detailed description of this function along with its historical origins and many applications are described in [57]. For $x \geq 0$, $W(x)$ is single valued (see Fig. 26).

II. Gronwall's Lemma

Let $[t_0, t_f] \mapsto y(t) \in \mathbb{R}$ be an integrable function that satisfies Gronwall's inequality [58],

$$y(t) \leq a(t) + \int_{t_0}^{t} b(s)y(s)\,\mathrm{d}s$$

where a and b are continuous, nonnegative, bounded functions with $t \mapsto a(t)$ nondecreasing over the interval, $[t_0, t_f]$; then

$$y(t) \leq a(t)e^{B(t)}$$

where

$$B(t) := \int_{t_0}^{t} b(s)\,\mathrm{d}s$$

For a proof of this lemma, please see [58].

III. Proof of Theorem 2

In the same spirit as Eq. (5), we have

$$\mathbf{x}^*(t^1) = \mathbf{x}_0 + \int_{t_0}^{t^1} \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t))\,\mathrm{d}t \tag{A2}$$

Subtracting Eqs. (5) and (A2) we get

$$\|\mathbf{x}^1(t^1) - \mathbf{x}^*(t^1)\| = \left\| \int_{t_0}^{t^1} \mathbf{f}(\mathbf{x}^1(t), \mathbf{u}^1(t))\,\mathrm{d}t \right.$$
$$\left. - \int_{t_0}^{t^1} \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t))\,\mathrm{d}t \right\| \leq \int_{t_0}^{t^1} \|\mathbf{f}(\mathbf{x}^1(t), \mathbf{u}^1(t))$$
$$- \mathbf{f}(\mathbf{x}^*(t), \mathbf{u}^*(t))\|\,\mathrm{d}t \leq \int_{t_0}^{t^1} (Lipf_x \|\mathbf{x}^1(t) - \mathbf{x}^*(t)\|$$
$$+ Lipf_u \|\mathbf{u}^1(t) - \mathbf{u}^*(t)\|)\,\mathrm{d}t \leq Lipf_u \|\mathbf{u}^1(\cdot) - \mathbf{u}^*(\cdot)\|_{L^\infty}(t^1 - t_0)$$
$$+ \int_{t_0}^{t^1} Lipf_x \|\mathbf{x}^1(t) - \mathbf{x}^*(t)\|\,\mathrm{d}t = \delta Lipf_x(t^1 - t_0)$$
$$+ \int_{t_0}^{t^1} Lipf_x \|\mathbf{x}^1(t) - \mathbf{x}^*(t)\|\,\mathrm{d}t \tag{A3}$$

where the equality in Eq. (A3) follows from the definition of δ given by Eq. (7). From Gronwall's Lemma, Eq. (A3) can be written as

$$\|\mathbf{x}^1(t^1) - \mathbf{x}^*(t^1)\| \leq \delta Lipf_x(t^1 - t_0)e^{Lipf_x(t^1 - t_0)} \tag{A4}$$

From the definition of the Lambert *W* function, it can be easily verified that for $y, z \in \mathbb{R}_+$, if $z \leq W(y)$ then $ze^z \leq y$; hence, from Eqs. (8) and (A4), we have $\|\mathbf{x}^1(t^1) - \mathbf{x}^*(t^1)\| \leq \delta r = \delta(\epsilon/\delta) = \epsilon$.

References

[1] Vinter, R. B., *Optimal Control*, Birkhäuser, Boston, MA, 2000, pp. 1–4, 30–39, 58, 60, 203–215, 435–491.
[2] Moyer, H. G., and Pinkham, G., "Several Trajectory Optimization Techniques, Part 2: Applications," *Computing Methods in Optimization Problems*, edited by A. V. Balakrishnan, and L. W. Neustadt, Academic Press, New York, 1964, pp. 91–105.
[3] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control*, Hemisphere, New York, 1975 (revised printing; original publication, 1969), pp. 66–69, 157.
[4] Bryson, A. E., *Dynamic Optimization*, Addison Wesley Longman, Inc., Reading, MA, 1999, pp. 125–130, 140–144, 183–185, 193–196, 381–401.
[5] Wall, B., and Conway, B. A., "Near-Optimal Low-Thrust Earth-Mars Trajectories via a Genetic Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, Sept.–Oct. 2005, pp. 1027–1031.
[6] 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.
[7] Hager, W. W., "Runge-Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik*, Vol. 87, No. 2, 2000, pp. 247–282.
[8] Ross, I. M., "Space Trajectory Optimization and L^1-Optimal Control Problems," *Modern Astrodynamics*, edited by P. Gurfil, Elsevier, St. Louis, MO, 2006, Chap. 6, pp. 155–188.
[9] Edelbaum, T. N., "Optimal Low-Thrust Rendezvous and Stationkeeping," *AIAA Journal*, Vol. 2, No. 12, 1971, pp. 1196–1201.

[10] Edelbaum, T. N., Sackett, L. L., and Malchow, H. L., "Optimal Low Thrust Geocentric Transfer," AIAA Paper 73-1074, Lake Tahoe, 1973.

[11] Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworths, London, 1963.

[12] http://trajectory.grc.nasa.gov/projects/lowthrust.shtml.

[13] Ross, I. M., "A Roadmap for Optimal Control: The Right Way to Commute," *Annals of the New York Academy of Sciences*, Vol. 1065, Jan. 2006, pp. 210–231.

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

[15] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Advances in Control and Design Series, SIAM, Philadelphia, PA, 2001.

[16] Ross, I. M., "A Historical Introduction to the Covector Mapping Principle," *Advances in the Astronautical Sciences*, Vol. 123, Univelt, San Diego, CA, 2006, pp. 1257–1278.

[17] Sussmann, H. J., and Williems, J. C., "300 Years of Optimal Control: From the Brachystochrone to the Maximum Principle," *IEEE Control Systems Magazine*, Vol. 17, June 1997, pp. 32–44.

[18] Mordukhovich, B. S., *Variational Analysis and Generalized Differentiation, 2: Applications*, Vol. 331, Grundlehren der Mathematischen Wissenschaften (Fundamental Principles of Mathematical Sciences) Series, Springer, Berlin, 2005, pp. 159–334.

[19] Gong, Q., Ross, I. M., Kang, W., and Fahroo, F., "On the Pseudospectral Covector Mapping Theorem for Nonlinear Optimal Control," *Proceedings of the 45th IEEE Conference on Decision and Control*, 13–15 Dec. 2006 (to be published).

[20] Paris, S. W., and Hargraves, C. R., OTIS 3.0 Manual, Boeing Space and Defense Group, Seattle, WA, 1996.

[21] Betts, J. T., and Huffman, W. P., *Sparse Optimal Control Software SOCS*, Mathematics and Engineering Analysis Technical Document MEA-LR-085, Boeing Information and Support Services, The Boeing Company, Seattle, WA, July 1997.

[22] Betts, J. T., and Erb, S. O., "Optimal Low Thrust Trajectories to the Moon," *SIAM Journal of Applied Dynamical Systems*, Vol. 2, No. 2, 2003, pp. 144–170.

[23] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131.

[24] Ross, I. M., Gong, Q., Fahroo, F., and Kang, W., "Practical Stabilization Through Real-Time Optimal Control," *2006 American Control Conference*, IEEE, Piscataway, NJ, 14–16 June 2006.

[25] Ross, I. M., Sekhavat, P., Fleming, A., and Gong, Q., "Pseudospectral Feedback Control: Foundations, Examples and Experimental Results," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 21–24 August 2006 (to be published).

[26] Ross, I. M., Gong, Q., and Sekhavat, P., "A Simple Technique for Low-Thrust High-Accuracy Trajectory Optimization," AAS Paper 06-150, 2006.

[27] Infeld, S. I., Josselyn, S. B., Murray, W., and Ross, I. M., "Design and Control of Libration Point Spacecraft Formations," *Journal of Guidance, Control, and Dynamics* (to be published).

[28] Ross, I. M., and D'Souza, C. N., "A Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, July–Aug. 2005, pp. 686–697.

[29] Mendy, P. B., "Multiple Satellite Trajectory Optimization," Astronautical Engineer Thesis, Department of Mechanical and Astronautical Engineering, Naval Postgraduate School, Monterey, CA, Dec. 2004.

[30] Infeld, S. I., "Optimization of Mission Design for Constrained Libration Point Space Missions," Ph.D. Dissertation, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, Jan. 2006.

[31] Trefethen, L. N., *Spectral Methods in MATLAB*, SIAM, Philadelphia, PA, 2000.

[32] Polak, E., "An Historical Survey of Computational Methods in Optimal Control," *SIAM Review* Vol. 15, No. 2, April 1973, pp. 553–584.

[33] Ferris, M. C., and Munson, T. S., "Interior-Point Methods For Massive Support Vector Machines," *SIAM Journal of Optimization*, Vol. 13, No. 3, 2003, pp. 783–804.

[34] Ross, I. M., and Fahroo, F., "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Sciences*, Vol. 295, Springer–Verlag, New York, 2003, pp. 327–342.

[35] Ross, I. M., "User's Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems," Tomlab Optimization Inc., TR 04-01.0, Feb. 2004.

[36] Hager, W. W., "Numerical Analysis in Optimal Control," *International Series of Numererical Mathematics*, edited by K.-H. Hoffmann, I. Lasiecka, G. Leugering, J. Sprekels, and F. Troeltzsch, Vol. 139, Birkhäuser, Basel, Switzerland, 2001, pp. 83–93.

[37] Betts, J. T., Biehn, N., and Campbell, S. L., "Convergence of Nonconvergent IRK Discretizations of Optimal Control Problems with State Inequality Constraints," *SIAM Journal of Scientific Computation* Vol. 23, No. 6, 2002, pp. 1981–2007.

[38] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342.

[39] Dickmanns, E. D., and Well, K. H., "Approximate Solution of Optimal Control Problems Using Third-Order Hermite Polynomial Functions," *Proceedings of the 6th Technical Conference on Optimization Techniques*, IFIP-TC7, Springer–Verlag, New York, 1975.

[40] Butcher, J. C., *The Numerical Analysis of Ordinary Differential Equations*, Wiley, New York, 1987.

[41] Gong, Q., Ross, I. M., Kang, W., and Fahroo, F., "Connections Between the Covector Mapping Theorem and Convergence of Pseudospectral Methods for Optimal Control," *Computational Optimization and Applications: An International Journal* (to be published), Springer Science, The Netherlands.

[42] Gong, Q., Kang, W., and Ross, I. M., "A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems," *IEEE Transactions on Automatic Control*, Vol. 51, No. 7, July 2006, pp. 1115–1129.

[43] Fahroo, F., and Ross, I. M., "Second Look at Approximating Differential Inclusions," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, Jan.–Feb. 2001, pp. 131–133.

[44] Fahroo, F., and Ross, I. M., "Direct Trajectory Optimization by a Chebychev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, Jan.–Feb. 2002, pp. 160–166.

[45] Ross, I. M., and Fahroo, F., "Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems," *IEEE Conference on Decision and Control*, IEEE, Piscataway, NJ, Dec. 2002.

[46] Melton, R. G., "Comparison of Direct Optimization Methods Applied to Solar Sail Problems," AIAA Paper 2002-4728, 2002.

[47] Lu, P., Sun, H., and Tsai, B., "Closed-Loop Endoatmospheric Ascent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 283–294.

[48] Young, L. C., *Lectures on the Calculus of Variations and Optimal Control Theory*, Saunders, Philadelphia, PA, 1969, pp. 291–292.

[49] Fuller, A. T., "Study of an Optimum Nonlinear Control System," *Journal of Electronics and Control*, Vol. 15, No. 1, 1963, pp. 63–71.

[50] Sauer, C. G., "Solar Electric Propulsion Performance for Medlite and Delta Class Planetary Missions," AAS Paper 97-726, 1997.

[51] Mengali, G., and Quarta, A. A., "Fuel-Optimal, Power-Limited Rendezvous with Variable Thruster Efficiency," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 6, Nov.–Dec. 2005, pp. 1194–1199.

[52] Williams, S. N., and Coverstone-Carroll, V., "Mars Missions Using Solar Electric Propulsion," *Journal of Spacecraft and Rockets*, Vol. 37, No. 5, 2000, pp. 71–77.

[53] Croley, P. A., "Reachable Sets for Multiple Asteroid Sample Return Missions," Astronautical Engineer's Degree Thesis, Naval Postgraduate School, Monterey, CA, June 2005.

[54] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M., *Nonlinear Programming: Theory and Algorithms*, Wiley, New York, 2006, pp. 23–24.

[55] Josselyn, S. B., "Optimization of Low-Thrust Trajectories with Terminal Aerocapture," Astronautical Engineer's Degree Thesis, Naval Postgraduate School, Monterey, CA, June 2003.

[56] Sovey, J. S., Rawlin, V. K., and Patterson, M. J., "Ion Propulsion Development Projects in U.S.: Space Electric Rocket Test 1 to Deep Space 1," *Journal of Propulsion and Power*, Vol. 17, No. 3, May–June 2001, pp. 517–526.

[57] Corless, R. M., Gonnet, G. H., Hare, D. E. G., Jeffrey, D. J., and Knuth, D. E., "On the Lambert *W* Function," *Advances in Computational Mathematics*, Vol. 5, No. 1, 1996, pp. 329–359.

[58] Zabczyk, J., *Mathematical Control Theory: An Introduction*, Birkhäuser, Boston, 1992, pp. 92–93.