JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 33, No. 4, July–August 2010

Discrete Pseudocontrol Sets for Optimal Control Problems

Yuri Ulybyshev*
Rocket-Space Corporation "Energia," 141070, Korolev, Moscow Region, Russia

DOI: 10.2514/1.47315

A method of optimal control problem solutions based on a new concept of pseudocontrol sets is presented. This approach combines large-scale linear programming algorithms with the well-known discretization of the continuous system dynamics on small segments and uses discrete pseudocontrol sets, which are considered independently for each segment. Every set is expressed as a mesh approximation of an admissible control space. The method is associated with significant increases in the number of decision variables and requires introducing artificial variables or pseudovariables. Terminal conditions are presented as a linear matrix equation. An extension of the matrix equation for the sums of the pseudocontrols is used to transform the problem into a linear programming form. Interior-point inequality constraints are represented as a linear matrix inequality. The resulting linear programming form is characterized by matrices that are very large and sparse. The number of decision variables is on the order of tens of thousands. In modern linear programming, there are effective interior-point algorithms to solve such problems. A minimum path-planning problem with nonlinear constraints, reentry trajectory optimization with maximum cross range, and maximum-radius orbit transfer are considered as application examples. The results of the last example are almost coincident with known solutions using other methods.

Nomenclature

$\mathbf{A}$	=	matrix for inequality constraints, Eq. (17)
$\mathbf{A}_e$	=	matrix for equality constraints, Eq. (15a)
a	=	semimajor axis
a_o	=	initial thrust acceleration
b	=	function of constraints
$\mathbf{b}$	=	vector of inequality constraints, Eq. (17)
C_L, C_D	=	lift and drag aerodynamic coefficients, Eq. (33)
D	=	drag acceleration
E	=	specific energy, Eq. (31)
$\mathbf{F}$	=	boundary condition function (2)
$\mathbf{f}$	=	state function (1)
h	=	altitude
h_e	=	energy integral
i	=	segment number
J	=	performance index, Eq. (23)
j	=	index of pseudocontrol vector
k	=	quantity of pseudocontrol vectors at each segment
L	=	lift acceleration
L_{ML}	=	mean longitude
m	=	number of boundary conditions
m_o	=	initial mass
m_s	=	spacecraft mass
$\dot{m}$	=	mass flow rate
n	=	quantity of segments
$\mathbf{O}$	=	zero string
$\mathbf{P_f}$	=	vector of boundary conditions, Eq. (2)
p	=	orbit parameter
p_1, p_2	=	elements of eccentricity vector, Eq. (43)
r	=	magnitude of radius vector
r_e	=	Earth's radius
r_x, r_y	=	coordinates
S	=	function of interior-point inequality constraint, Eq. (4)
S_m	=	reference area
$s_i^{(j)}$	=	element of matrix $\mathbf{A}$
T	=	thrust magnitude
t	=	time
$\mathbf{q}$	=	weight vector, Eq. (21)
$q_i^{(j)}$	=	weight coefficients, Eq. (21)
U	=	scalar control
$\mathbf{U}$	=	control vector
u	=	radial velocity, Eq. (39)
V	=	magnitude of vehicle velocity
v	=	tangential velocity
$\mathbf{X}$	=	vector of decision variables, Eq. (10)
$x_i^{(j)}$	=	decision variable
$\mathbf{Y}$	=	state vector
α	=	angle of attack
ΔE	=	change of specific energy on ith segment
Δr_x	=	change of r_x on ith segment
Δt_i	=	duration of ith segment
Θ	=	flight-path angle
θ	=	angular displacement, Eq. (39)

Yuri Ulybyshev graduated from the Bauman Moscow High Technical School (now the Bauman Moscow State Technical University or BMSTU) in Flight Dynamics and Control. Since 1977, he has worked in the Space Ballistics Department of NPO Energia (now the Rocket-Space Corporation "Energia") and is presently the Head of the department. He has been involved in various space projects, such as the Space Shuttle Buran, the launch vehicle Energia, the spacecraft Soyz, the geostationary satellite Yamal, the Mir space stations and the International Space Station, the Sea Launch, and others. In 1990, he received his Ph.D. in technical sciences, and in 2002, he received his Dr.Sc. in technical sciences (in the BMSTU). He has been recognized in *Who's Who in the World* (since 1998) and in *Who's Who in Science and Engineering* (since 2005). He is a coeditor and an author of the engineering encyclopedia of the astronautics (in Russian). Currently, he also works as an Invited Professor in the BMSTU. His research interests include astrodynamics, spacecraft design, and control theory. He is a Senior Member of the AIAA.

Presented as Paper 5788 at the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, 10–13 August 2009; received 22 September 2009; revision received 22 March 2010; accepted for publication 22 March 2010.

*yuri.ulybyshev@rsce.ru; yuri.ulybyshev@gmail.com.

λ = latitude
μ = gravitational parameter for the Earth or the sun
μ_U = coefficient for control refinement, Eq. (38)
ν = number of segments with inequality constraints
ξ = dimension of control space
ρ = atmospheric density
σ = bank angle
Φ = forbidden area
ϕ = angle of thrust direction
φ = longitude
Ψ = heading angle
Ω = admissible subspace for control vector
ω_π = argument of perigee

I. Introduction

OPTIMAL control methods have been mainly of two types: indirect and direct techniques or their combinations [1]. The indirect methods solve the classical optimal control problem by obtaining the solution to the corresponding two-point boundary-value problem based on the Pontryagin maximum principle [2]. The boundary-value problem can be solved numerically, although it becomes a very difficult task for realistic problems since, in the general case, a good initial guess for unknown initial costate variables is usually not available. The direct methods discretize the original problem, transforming it into a parameter optimization problem, which is then solved using, as a rule, nonlinear programming methods [1]. The methods are attractive because explicit consideration of the necessary optimal conditions (adjoint equations and transversality conditions) are not required. A general review of the methods for space trajectory optimization is presented by Betts [3].

Linear programming [4] represents one of the well-known optimization methods successfully used to solve many complex application problems in engineering, economics, and operations research. Linear programming for linear optimal control problems is proposed by Dantzig [5]. For this case, the control can be found as a linear combination from a set of feasible control vectors that satisfy the boundary conditions. An explicit application of linear programming for spacecraft trajectory optimization is difficult even for linear motion models. As a rule, it is related to the nonlinearity of the performance index, which is usually the total characteristic velocity. Ulybyshev and Sokolov [6] have developed a method for optimization of many-revolution low-thrust maneuvers in the vicinity of the geostationary orbit. The state variables are longitude, longitude drift rate, and eccentricity. The proposed mathematical model introduces pseudomaneuvers with either positive or negative transverse directions of the thrust vector for every trajectory segment (half a revolution) and transforms the performance index to a linear form. This makes it possible to state the problem in terms of classical linear programming with a number of decision variables equal to quadruple the number of revolutions in the orbit transfer. In a sense, similar techniques with an extension of control variables and performance index linearization is applied to control allocation [7,8] and to on–off minimum-time control [9,10]. The majority of the methods use the well-known simplex method with the total number of decision variables on the order of tens or hundreds. The mixed-integer linear programming approach with trajectory discretization is proposed for spacecraft trajectory planning with avoidance constraints [11]. The spacecraft are in close proximity and a linearized model of relative motion is used. The constraints can be transformed into a mixed-integer form by introducing binary variables. A set of binary variables (0 or 1) are added to the problem for each pair of vehicles at each time step.

In the 1990s, linear programming underwent a revolution with the development of polynomial-time algorithms known as interior-point methods [12–14]. The search directions for these methods strike out into the interior of the polytope rather than skirting around the boundaries, as do the well-known simplex methods. Many studies now show that for large linear problems, the interior-point methods do better than classical simplex methods. Furthermore, unlike simplex-based algorithms that have difficulty with degenerate problems, interior-point methods are immune to degeneracy [15]. A general case of the optimal control problem can be considered using the interior-point algorithms.

The proposed method is based on two principles. The first one is a well-known discretization of the time or other independent variable, in the general case, on nonuniform segments. The key idea of the second one is based on a small-width grid approximation of the control space by a set of pseudocontrol vectors with an equality constraint for each segment. Similar methods using discrete sets of pseudoimpulses were developed by the author for spacecraft trajectory optimization [16–18]: maintenance of a 24-h elliptical orbit, coplanar and noncoplanar orbit transfers, various rendezvous trajectories for near-circular orbits (with high, medium, and low thrust), three-dimensional launch trajectory from the moon surface to a circumlunar orbit with constraints, optimization of lunar landing trajectories with safety descent profile, thrust level, and attitude constraints. These examples differ essentially from a general form of the optimal control problem since the control (i.e., spacecraft maneuvers) is usually present only on some parts of the spacecraft trajectory. By contrast, the general case of optimal control requires a continuous control. Such problems are considered in the paper. The contribution of the paper is an extension to the concept of discrete pseudoimpulse sets for the solution of a more general optimal control problem and demonstration that the proposed concept of pseudocontrol sets, in spite of a high order, is a good candidate to solve optimal control problems.

II. Optimal Control Problem

A. Problem Formulation

We consider optimal control problems, where the initial time t_0 and final time t_f are specified. A general optimal control problem can be formulated as follows. Find an admissible optimal control vector $\mathbf{U}(t) \in \Omega \subseteq R^\xi$, such that the dynamic system described by the differential equation

$$\frac{\mathrm{d}\mathbf{Y}}{\mathrm{d}t} = \mathbf{f}(\mathbf{Y}, \mathbf{U}, t) \tag{1}$$

is transferred from an initial state $\mathbf{Y}(t_0)$ into a final state $\mathbf{Y}(t_f)$, satisfying a terminal constraint

$$\mathbf{F}(\mathbf{Y}, \mathbf{U}, t_f) = \mathbf{P}_f \tag{2}$$

such that the performance index

$$J = \int_{t_0}^{t_f} q(\mathbf{Y}, \mathbf{U}, t)\,\mathrm{d}t \tag{3}$$

is minimized. Here, $\mathbf{F}$ is an m-dimensional vector function of the state vector at the final time and $\mathbf{P}_f$ is an m-dimensional specified vector of the terminal conditions.

Remark 1: An arbitrary monotonically changing variable can be used as the independent variable instead of time. A final value of the variable is specified.

Remark 2: Optimal control problems are often required to satisfy not only terminal conditions but also some specific constraints for interior points in the form of an inequality,

$$S(\mathbf{Y}, t) - b(t) \le 0 \quad \text{for} \quad t_0 \le t_b \le t \le t_e \le t_f \tag{4}$$

where $b(t)$ is a specified function, and t_b and t_e are the beginning and end times for a trajectory.

B. Trajectory Discretization

Introduce a set of segments as the partition $[t_0, t_1, t_2, \ldots, t_n]$, with $t_0 = 0$ and $t_n = t_f$, and $t_0 < t_1 < t_2 \cdots < t_n$, and let $\Delta t_i = t_{i+1} - t_i$ be a small time interval for $i = 1, 2, \ldots, n$. The mesh points t_i are referred to as nodes, the intervals $\Delta t_i = [t_{i+1}, t_i]$ are referred to as trajectory segments. In the general case, the segments can be nonuniform. Suppose that approximate values of the state vectors at

the nodes $\mathbf{Y}(t_i)$ are known, then for a constant control at each segment, we can write

$$\mathbf{P}_f = \mathbf{F}(\mathbf{Y}_0, \mathbf{U}_0, 0) + \sum_{i=1}^{i=n} \Delta\mathbf{F}(\mathbf{Y}_i, \mathbf{U}_i, t_i, \Delta t_i) = \mathbf{F}_0 + \sum_{i=1}^{i=n} \frac{\partial \mathbf{F}(\mathbf{Y}_i, \mathbf{U}_i, t_i)}{\partial t} \cdot \Delta t_i \tag{5}$$

where $\mathbf{F}(\mathbf{Y}_0, \mathbf{U}_0, 0)$ is the terminal function (2) at the initial point, $\Delta\mathbf{F}$ is a possible change on ith segment, and $\partial\mathbf{F}/\partial t$ is a partial derivative. Note that the segment duration Δt_i may be defined in an implicit form.

III. Linear Programming Form

A. Discrete Approximation of Control Space

The simplest case of the continuous control is a scalar function. We consider an ith segment independent of all the other segments. Suppose that there is an optimal value $U_{i\mathrm{OPT}} \in \Omega \equiv [u_{\min}, u_{\max}]$ for the segment. Then,

$$\left.\frac{\partial J}{\partial U}\right|_{t=t_i, u=u_{i\mathrm{opt}}} \equiv 0 \tag{6}$$

and the performance index in a neighborhood of the optimal value varies with the square of a control change from the optimal value (see Fig. 1). We will address the very significant increases in the number of decision variables by introducing artificial variables or pseudovariables. For each segment, the possible values of the function can be present as a discrete set of pseudocontrols $U_i^{(j)} \in \Omega \equiv [U_{\min}, U_{\max}]$, where $j = \overline{1,k}$ is an index.

We can replace the optimal value by an approximating sum,

$$U_{i\mathrm{OPT}} = \sum_{j=1}^{j=k} x_i^{(j)} \cdot U_i^{(j)} \tag{7}$$

where

$$\sum_{j=1}^{j=k} x_i^{(j)} = 1 \tag{8}$$

$$0 \le x_i^{(j)} \le 1 \tag{9}$$

where $x_i^{(j)}$ is a decision variable.

It is evident that the optimal approximation of $U_{i\mathrm{OPT}}$ by the pseudocontrols is a sum of the two nearest-neighbor pseudocontrol values $x_i^{(j)} U_i^{(j)} \le U_{i\mathrm{OPT}} \le x_i^{(j+1)} U_i^{(j+1)}$. In a particular case, it can be one of the nearest-neighbor pseudocontrol values.

In a similar way, we consider a multidimensional case of the control. For each segment, a set of pseudocontrol vectors can be constructed using a multidimensional mesh (see Fig. 2). Similar to the scalar case, the sums of the pseudocontrol vectors should be constrained by Eqs. (8) and (9). The best approximation of the vector $\mathbf{U}_{i\mathrm{OPT}}$ is also a sum of nearest-neighbor pseudocontrol vectors or one nearest-neighbor pseudocontrol vector.

It should be noted that each segment can use distinct control areas $\mathbf{U}_i \in \Omega_i$ and corresponding meshes for pseudocontrol vectors.

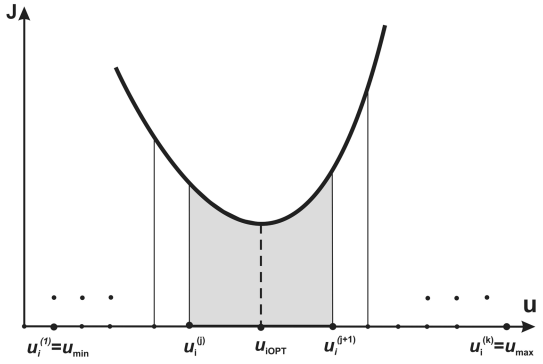

Fig. 1 Set of pseudocontrols for a segment.

B. Transformation into Linear Programming Form

Define a $(n \times k)$-dimensional vector of the decision variables:

$$\mathbf{X}^T = [x_1^{(1)}, x_1^{(2)}, \ldots x_1^{(k)}, x_2^{(1)}, x_2^{(2)}, \ldots x_2^{(k)}, \ldots x_n^{(k)}] \tag{10}$$

For the vector, according to the previous statements, the following linear matrix equality can be written:

$$\mathbf{A}_e^* \mathbf{X} = \mathbf{P}_d \tag{11}$$

where $\mathbf{A}_e^*$ is an $n(n \times k)$-dimensional matrix of the following form (all of the unspecified elements equal to zero):

$$\mathbf{A}_e^* = \underbrace{\left[\begin{array}{cccc} \underbrace{111\cdots1}_{k} & & & \\ & \underbrace{111\cdots1}_{k} & & \\ & & \cdots & \\ & & & \underbrace{111\cdots1}_{k} \end{array}\right]}_{n\times k} \Bigg\} n \tag{12}$$

and an n-dimensional vector

$$\mathbf{P}_d^T = [1, \quad 1, \quad 1\cdots1, \quad 1] \tag{13}$$

Equation (5) for terminal conditions can be presented as

$$\Delta\mathbf{P}_f = \mathbf{P}_f - \mathbf{F}(\mathbf{Y}_0, \mathbf{U}_0, 0) = \sum_{i=1}^{i=n}\sum_{j=1}^{j=k} x_i^{(j)} \Delta\mathbf{F}(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, t_i, \Delta t_i) = \sum_{i=1}^{i=n}\sum_{j=1}^{j=k} x_i^{(j)} \frac{\partial \mathbf{F}(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, t_i)}{\partial t} \cdot \Delta t_i \tag{14}$$

where $\mathbf{U}_i^{(j)}$is pseudocontrol vector. Now, one can combine Eqs. (5) and (11) to write

$$\mathbf{A}_\mathbf{e}\mathbf{X} = \mathbf{P} \tag{15a}$$

where $\mathbf{P}^T = [\mathbf{P}_d^T, \Delta\mathbf{P}_f^T]$ and $\mathbf{A}_\mathbf{e}$ is a $(n+m) \times (n \times k)$-dimensional matrix:

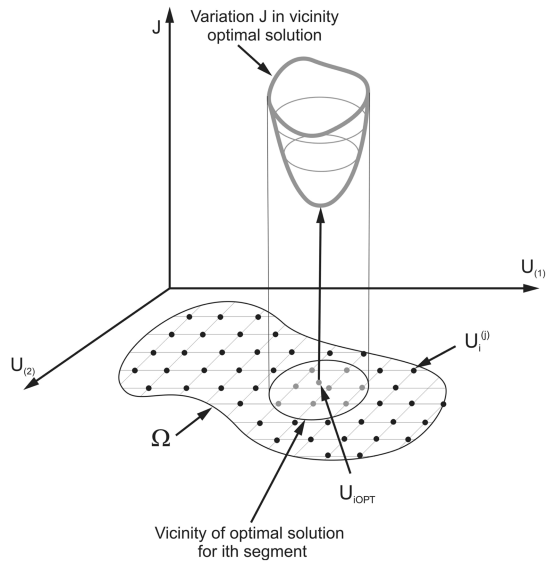

Fig. 2 Set of pseudocontrols for two-dimensional control vector.

$$\mathbf{A}_{\mathbf{e}} = \left[\underbrace{\frac{\partial \mathbf{F}(\mathbf{Y}_1, \mathbf{U}_1^{(1)}, t_1)}{\partial t} \cdot \Delta t_1 \quad \frac{\partial \mathbf{F}(\mathbf{Y}_1, \mathbf{U}_1^{(2)}, t_1)}{\partial t} \cdot \Delta t_1 \quad \cdots \quad \frac{\partial \mathbf{F}(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, t_i)}{\partial t} \cdot \Delta t_i \quad \cdots \quad \frac{\partial \mathbf{F}(\mathbf{Y}_n, \mathbf{U}_n^{(k)}, t_n)}{\partial t} \cdot \Delta t_n}_{\mathbf{A}_{\mathbf{e}}^*} \right] \tag{15b}$$

Each interior-points constraint [Eq. (4)] can be presented as a number of inequality constraints. The number of ν is equal to the quantity of the segments inside the interval of $t_0 \le t_b \le t \le t_e \le t_f$. Each inequality can be written as

$$S_l = S(\mathbf{Y}_0, \mathbf{U}_0, 0) + \sum_{i=1}^{i=l} \sum_{j=1}^{k} x_i^{(j)} \Delta S(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, t_i, \Delta t_i) = S_0 + \sum_{i=1}^{i=l} \sum_{j=1}^{k} x_i^{(j)} \frac{\partial S(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, t_i)}{\partial t} \cdot \Delta t_i \le b(t_i); \qquad l = \overline{1, \nu} \tag{16}$$

or for the inequalities in matrix form,

$$\mathbf{A}\mathbf{X} \le \mathbf{b} \tag{17}$$

where

$$\mathbf{b}^{\mathbf{T}} = [\, b(t_1) - S(\mathbf{Y}_0, \mathbf{U}_0, 0), \quad b(t_2) - S(\mathbf{Y}_0, \mathbf{U}_0, 0), \quad \cdots \quad b(t_n) - S(\mathbf{Y}_0, \mathbf{U}_0, 0) \,] \tag{18}$$

$$\mathbf{A} = \begin{bmatrix} \underbrace{s_1^{(1)} \ \ldots \ s_1^{(k)}}_{k} & \cdots & \underbrace{s_\beta^{(1)} \ \ldots \ s_\beta^{(k)}}_{k} & \mathbf{O}_k & \mathbf{O}_k & \cdots & \mathbf{O}_k \\ \underbrace{s_2^{(1)} \ \ldots \ s_2^{(k)}}_{k} & \cdots & \underbrace{s_\beta^{(1)} \ \ldots \ s_\beta^{(k)}}_{k} & \underbrace{s_{\beta+1}^{(1)} \ \ldots \ s_{\beta+1}^{(k)}}_{k} & \mathbf{O}_k & \cdots & \mathbf{O}_k \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \underbrace{s_1^{(1)} \ \ldots \ s_{1\nu}^{(k)}}_{k} & \cdots & \underbrace{s_\beta^{(1)} \ \ldots \ s_\beta^{(k)}}_{k} & \underbrace{s_{\beta+1}^{(1)} \ \ldots \ s_{\beta+1}^{(k)}}_{k} & \cdots & \underbrace{s_{\beta+\nu}^{(1)} \ \ldots \ s_{\beta+\nu}^{(k)}}_{k} & \mathbf{O}_{kx(n-\beta-\nu)} \end{bmatrix} \tag{19}$$

where $\mathbf{O}$ is a zero string and

$$s_i^{(j)} = \frac{\partial S(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, t_i)}{\partial t} \cdot \Delta t_i \tag{20}$$

Note that interior-point equality constraints and other type constraints can be expressed in a similar way as a linear matrix equality and/or inequality. Some details are described in [18].

Introduce a $(n \times k)$-dimensional vector of weight coefficients as

$$\mathbf{q}^T = [q_1^{(1)}, q_1^{(2)}, \ldots, q_1^{(k)}, \ldots, q_i^{(j)}, \ldots, q_n^{(k-1)}, q_n^{(k)}] \tag{21}$$

where

$$q_i^{(j)} = q(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, t_i) \tag{22}$$

Then, the performance index [Eq. (3)] can be written as

$$J = \min(\mathbf{q}^{\mathbf{T}} \cdot \mathbf{X}) \tag{23}$$

As the final result, we have a classical linear programming problem with constraints of a linear matrix equality and two inequalities given by Eqs. (9), (14), and (17), respectively.

C. Computational and Qualitative Aspects of the Proposed Methods

The proposed linear programming form is a large-scale problem. As an example, for a trajectory with 100 segments and 1000 pseudocontrol vectors at each segment, the number of decision variables is 100,000. For $m = 2$, the matrix $\mathbf{A}_{\mathbf{e}}$ dimension is $(1000 + 2) \times 100,000$. But it is a sparse matrix with a very low number of nonzero elements ($\sim$0.1%). Modern scientific software, such as MATLAB® [19], contains effective algorithms for sparse matrix computations, including large-scale linear programming algorithms. A typical solution process requires less than 1–3 min of computation time using a Pentium IV processor.

The segments in the linear programming form are formally considered independent of each other. Therefore, an additional postprocessing and validation are required for the linear programming solutions. It is necessary to find all of the nonzero decision variables (i.e., more than a tolerance of $\sim 10^{-3}$–10^{-5}) for the segments. For a scalar control function, no more than two pseudocontrols can be presented at each segment. The two pseudocontrols should be adjacent, and the optimal value is $U_{i\mathrm{OPT}} = x_i^{(j)} \cdot U_i^{(j)} + x_i^{(j+1)} \cdot U_i^{(j+1)}$. A simple example of such postprocessing algorithm is given in Appendix A. For a multidimensional case and several nonzero decision variables belonging to a segment, the optimal control vector should be computed from the sum of the corresponding pseudocontrols vectors. The vectors must be nearest-neighbor pseudocontrol vectors to the optimal value. Figure 3 is a schematic illustration of this process. But experience shows that for a small-width grid approximation of the control space, almost all of the linear programming solutions have only one pseudocontrol vector at each segment. The presence of two or more pseudocontrol vectors for a segment is rare.

There are some important points about the mathematical implementation of the problem. For the linear control problems, the partial derivatives in Eq. (14) are known. For nonlinear control problems, the partial derivatives are usually not known a priori. For this case, we can use an iterative technique with a refinement of the partial derivatives at each iteration. For the first iteration, we define an initial guess for the system motion. Based on the first linear programming solution, the derivatives are refined for the second iteration, etc., to obtain an accessible solution. Contrary to the linear programming method, the optimal control techniques based on an iterative solution of a two-point boundary-value problem for the state and adjoint variables are difficult to apply. The main difficulty with these methods is getting started (i.e., finding a first estimation of the unspecified conditions for the state and adjoint variables [1]).

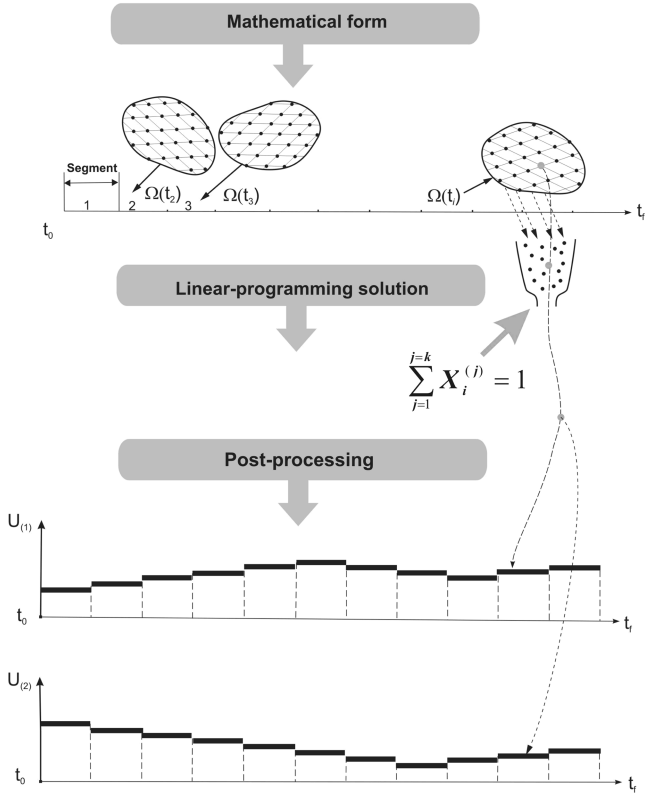

Fig. 3 Postprocessing of the linear programming solutions.

Moreover, the adjoint variables do not have a physical meaning; thus, it can be difficult to find a reasonable initial guess for them. Generally, that search for an initial guess for the state variables (i.e., for a trajectory) is a simpler problem than the similar problem for the adjoint variables. An application possibility of the presented method for nonlinear problems needs to be studied, but there are some examples of nonlinear solutions for optimal spacecraft trajectories [16]. Examples of a nonlinear optimal control problem will be considered below.

Since the segments are formally considered independent of each other, there is a major restriction for the method application. The relation between the decision variables and the partial derivatives should be in an explicit form. As an example, suppose that there is a terminal constraint for a coordinate but there is an explicit control dependence only for the second derivative of the coordinate. For this case, the Taylor series can be used (see example B in the next section).

The interior-point methods are not only highly efficient algorithms for large-scale linear programming, but they are immune to degeneracy [15]. Therefore, the absence of a solution means, most likely, that the problem formulation with terminal conditions and constraints is degenerate.

By contrast with most traditional direct methods using nonlinear programming [20], the proposed technique has some attractive properties. The nonlinear programming methods have difficulties with the presence of discontinuous behavior since they assume that the objective and constraint functions are continuous and differentiable [20]. A similar statement is true for a bang–bang control. The proposed technique does not require continuity of the state variables and/or control (see example B in the next section). Another attractive property of the proposed technique is the explicit form for the interior-point equality and inequality constraints. Examples are given in the next section (example A), and in [17,18]. For nonlinear programming methods, there are great difficulties related with inequality constraints that can be active or inactive at each instant in time (i.e., the time domain is partitioned into constrained and unconstrained subarcs). The number of constrained subarcs present in the optimal solution is not known a priori, and the location of the junction points when the transition from constrained to unconstrained (and vice versa) occurs is unknown [20].

IV. Application Examples

A. Minimum Path Planning with Nonlinear Constraints

Planar motion of a vehicle with constant velocity can be presented as

$$\begin{cases} \frac{dr_x}{dt} = V\cos[\Theta(t)] \\ \frac{dr_y}{dt} = V\sin[\Theta(t)] \end{cases} \tag{24}$$

where r_x and r_y are the coordinates, V is the velocity, and $\Theta(t)$ is the control variable (flight-path angle).

Consider the following optimal control problem: find the minimum path or minimum-time trajectory from an initial point (r_{x0}, r_{y0}) to a terminal point (r_{xf}, r_{yf}) with constraints on the trajectory $[r_x(t), r_y(t)] \notin \Phi$, where Φ is a forbidden area (in the general case, it is a multiconnected nonconvex area). Suppose that either of two coordinates is a monotonic variable for the trajectory. Undoubtedly, this assumption is applicable to a part of the path-planning problems but it makes possible a transformation of the problem into the linear programming form. Let r_x be the monotonic or independent variable. Then, we can rewrite Eq. (24) as

$$\frac{dr_y}{dr_x} = \tan[\Theta(r_x)] \tag{25}$$

The performance index is

$$J = \int_{r_{x0}}^{r_{xf}} \frac{1}{V\cos[\Theta(r_x)]}\, dr_x \tag{26}$$

We assume that the trajectory discretized into n segments with k pseudocontrols $\Theta_i^{(j)}$ in each segment. Then, for the partial derivatives [Eq. (14)] and performance index elements [Eq. (22)], we can write

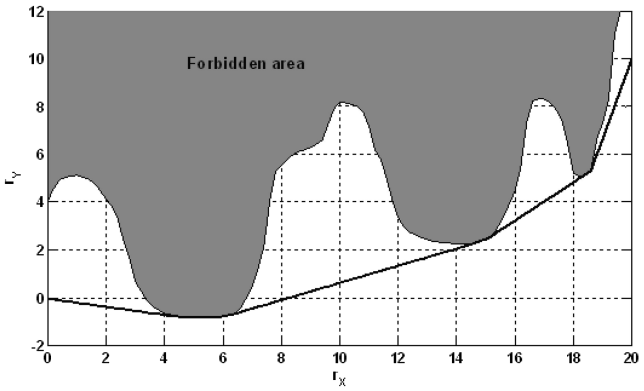

Fig. 4 Trajectory for example 1.

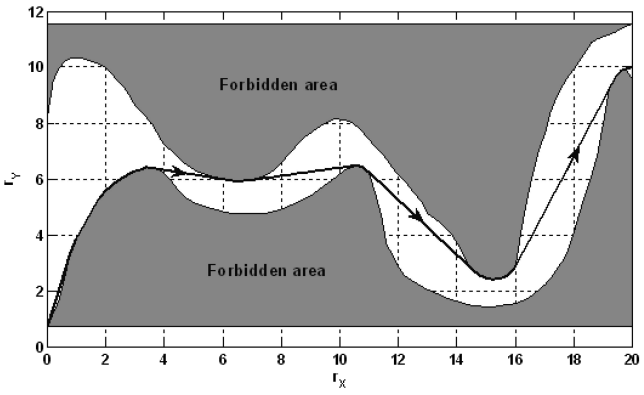

Fig. 5 Trajectory for example 2.

$$\Delta \mathbf{F}_i^{(j)} = \Delta r_x \tan[\Theta_i^{(j)}] \tag{27}$$

$$q_i^{(j)} = \frac{\Delta r_x}{V \cos[\Theta_i^{(j)}]} \tag{28}$$

where Δr_x is a discretization step. The forbidden area Φ can be presented as inequalities (17) for the second coordinate r_{yi} at the corresponding segments.

As examples, we consider two trajectories from $(r_{x0}, r_{y0}) = (0, 0)$ to $(r_{xf}, r_{yf}) = (20, 10)$ with constraints, which are shown in Figs. 4 and 5. The constraints can be expressed in the following mathematical form:

$$S = r_y - b_{ry}(r_x) \le 0 \tag{29}$$

for an upper bound and

$$S = -r_y + b_{ry}(r_x) \le 0 \tag{30}$$

for a lower bound.

The optimal trajectories are also depicted in the figures. The control variables histories are presented in Fig. 6. The results are given for 101 uniformly distributed segments with $\Delta r_x = 0.2$ and 176 uniformly distributed pseudocontrols with $\Delta\Theta = \Theta_i^{(j+1)} - \Theta_i^{(j)} = 1^\circ$. For the trajectories, there are two arc types. The first is a straight line and the second is a tracking of forbidden area boundaries.

B. Reentry Trajectory Optimization with Maximum Cross Range

For reentry trajectories of an unpowered spacecraft, energy is a more appropriate independent variable than time [21]. The specific energy E is given by

$$E = \frac{1}{2}V^2 - [(\mu/r) - (\mu/r_e)] \tag{31}$$

where r is the distance from the center of the Earth, V is the velocity magnitude, μ is the gravitational parameter for the Earth, and r_e is the Earth's radius. Using $\dot{E} = -VD$ and denoting $\mathrm{d}(\bullet)/\mathrm{d}E$ as $(\bullet)'$, the motion equations [21] for a nonrotating spherical Earth can be written as

$$\begin{cases} r' = -\sin\Theta/D \\ \varphi' = -\frac{\cos\Theta\cos\Psi}{Dr\cos\lambda} \\ \lambda' = -\frac{\cos\Theta\sin\Psi}{Dr} \\ V' = \frac{D+g\sin\Theta}{DV} \\ \Theta' = -\frac{\{L\cos\sigma-[g-(V^2/r)]\cos\Theta\}}{DV^2} \\ \Psi' = -\frac{1}{DV^2}\left[\frac{L\sin\sigma}{\cos\Theta} - \frac{V^2\cos\Theta\cos\Psi\tan\lambda}{r}\right] \end{cases} \tag{32}$$

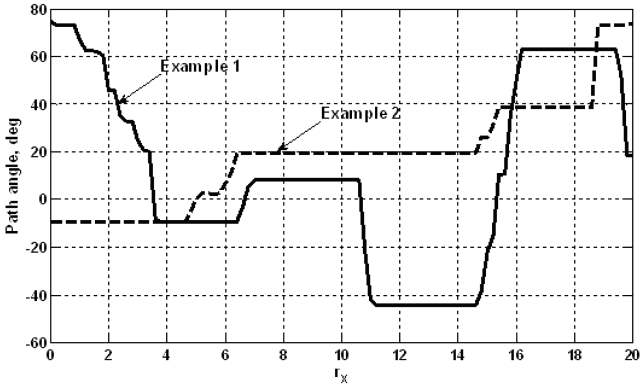

Fig. 6 Control histories.

where φ is the longitude, λ is the latitude, Θ is the flight-path angle, Ψ is the heading angle, σ is the bank angle, and

$$D = \frac{1}{2m_s}\rho(h)S_m V^2 C_D(\alpha); \qquad L = \frac{1}{2m_s}\rho(h)S_m V^2 C_L(\alpha) \tag{33}$$

where $\rho(h)$ is the atmospheric density as a function of altitude $(h = r - r_e)$, and $C_L(\alpha)$ and $C_D(\alpha)$ are the lift and drag coefficients as functions of the angle of attack α.

The optimal control problem is as follows: to determine the control vector $\mathbf{U}^T(E) = [\alpha(E), \sigma(E)]^T$ that steers the vehicle on a feasible trajectory passing through the final point with desired r_f (or altitude) with a maximum cross range.

The problem is nonlinear and the partial derivatives in Eqs. (15) and (16) are unknown a priori. For this case, we can use an iterative technique with a refinement of the partial derivatives at each iteration. Suppose that there is an initial control law $\mathbf{U}_{(0)}(E)$ with the corresponding reference trajectory. For the specific energy as independent variable, we have specified values for their initial and final values E_0 and E_f, which are computed from Eq. (31). The trajectory can be divided into n segments (between E_0 and E_f with $\Delta E = [E_0 - E_f]/n$). Each of them includes k pseudocontrol vectors $\mathbf{U}_i^{(j)}$. The velocities r' and λ' are independent of the bank angle. Then, the functions from Eqs. (15) and (16) can be computed along the reference trajectory as parts of Taylor series:

$$\Delta F(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, \Delta E) = r'(\mathbf{Y}_i, \alpha_i^{(j)})\Delta E + \tfrac{1}{2}r''(\mathbf{Y}_i, \alpha_i^{(j)}, \sigma_i^{(j)})\Delta E^2 \tag{34}$$

$$\Delta S(\mathbf{Y}_i, \mathbf{U}_i^{(j)}, \Delta E) = -[\lambda'(\mathbf{Y}_i, \alpha_i^{(j)})\Delta E + \tfrac{1}{2}\lambda''(\mathbf{Y}_i, \alpha_i^{(j)}, \sigma_i^{(j)})\Delta E^2] \tag{35}$$

where

$$r''(\mathbf{Y}_i, \mathbf{U}_i^{(j)}) = -\frac{\cos\Theta(\mathbf{Y}_i)\cdot\Theta'(\mathbf{Y}_i, \alpha_i^{(j)}, \sigma_i^{(j)})}{D(\mathbf{Y}_i, \alpha_i^{(j)})} \tag{36}$$

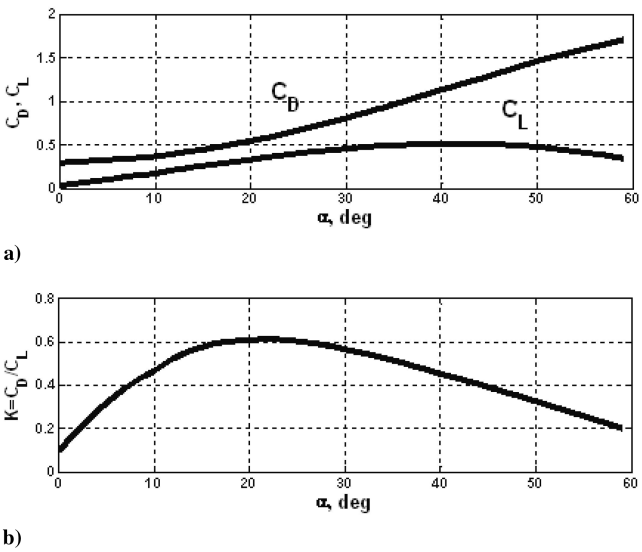

Fig. 7 Aerodynamic coefficients.

$$\lambda''(\mathbf{Y}_i, \mathbf{U}_i^{(j)}) = \frac{\sin\Theta(\mathbf{Y}_i)\cdot\sin\Psi(\mathbf{Y}_i)\cdot\Theta'(\mathbf{Y}_i,\alpha_i^{(j)},\sigma_i^{(j)}) - \cos\Theta(\mathbf{Y}_i)\cdot\cos\Psi(\mathbf{Y}_i)\cdot\Psi'(\mathbf{Y}_i,\alpha_i^{(j)},\sigma_i^{(j)})}{r(\mathbf{Y}_i)\cdot D(\mathbf{Y}_i,\alpha_i^{(j)})} - \frac{\cos\Theta(\mathbf{Y}_i)\cdot\sin\varphi(\mathbf{Y}_i)\cdot r'(\mathbf{Y}_i,\alpha_i^{(j)})}{r(\mathbf{Y}_i)^2\cdot D(\mathbf{Y}_i,\alpha_i^{(j)})} \quad (37)$$

Based on the linear programming solution and postprocessing, we compute a control law $\mathbf{U}^*_{(1)}(E)$. The new control can be computed as a combination of the law and control from the previous iteration:

$$\mathbf{U}_{(1)}(E) = \mu_U \mathbf{U}^*_{(1)}(E) + (1-\mu_U)\mathbf{U}_{(0)}(E) \quad (38)$$

where $0 < \mu_U \le 1$ is a coefficient. Further, we compute a new trajectory passing through r_f with a new value of the final energy E_f and, respectively, $\Delta E = [E_0 - E_f]/n$. The partial derivatives (15) and (16) are refined for the second iteration, etc., to obtain an acceptable solution. There are well established methods for detecting whether an iterative process has converged. The most common way to measure progress is to assign some merit function [20]. An obvious merit function, which was used in this case, is $J[\mathbf{U}(E)]$. But for the general case, the problem needs study. It should be noted that, for $\mu_U = 1$, the iterative process may be divergent if there is a significant disagreement between the initial trajectory and trajectory with refined control and, respectively, between the corresponding partial derivatives. Experience shows that, for the reentry problem as presented here, $\mu_U \cong 0.1 - 0.4$ is more suitable.

As an example, we consider a low-lift vehicle with a mass of $m_s = 10,000$ kg, a reference area of $S_m = 10$ m^2, a maximum lift-to-drag ratio of $K = C_L/C_D = 0.6$, and the lift and drag coefficients as shown in Fig. 7.

The initial state vector is

$$\mathbf{Y}_0^T = [r_0 = 6438 \text{ km}(h = 60 \text{ km}); \varphi_0 = 0; \lambda_0 = 0; V_0 = 6.0 \text{ km/s}; \Theta_0 = -0.25^\circ; \Psi_0 = 0]^T$$

The terminal condition is $r_f = 6396$ km$(h = 28$ km$)$. The trajectory is divided into $n = 100$ segments, each of them includes $k = 1800$ pseudocontrol vectors of $\mathbf{U}_i^{(j)}$. The vectors are presented in a mesh grid with (40×45) elements for angles of attack $[\alpha = 10 - 30^\circ, (\Delta\alpha = 0.5^\circ)]$ and bank angles $[\sigma = 30\text{–}75^\circ, (\Delta\sigma = 1^\circ)]$. Therefore, the number of decision variables is $(100 \times 1800) = 180,000$. For the initial reference trajectory, we use a constant control law:$\alpha = 15^\circ$ and $\sigma = 35^\circ$.

The optimization results for six iterations ($\mu_U = 0.2$) are given in Table 1 and Figs. 8–12 (initial reference trajectory: dashed line, trajectories at iterations: thin solid lines, and optimal trajectory: bold solid line). For the initial trajectory with constant control, the performance index is $J_{\text{IN}} = 129.03$ km, and for the optimal solution, $J = 165.43$ km.

Figures 8 and 9 show the trajectories in the vertical and horizontal planes (the longitudinal range is $r_e\varphi$, and the cross range is $r_e\lambda$). The histories of the angle of attack and bank angle are shown in Figs. 10 and 11, respectively. The flight-path and heading angle histories are depicted in Fig. 12. It is interesting to note that, for each iteration, there are distinct values of the final time, energy, and velocity. For the angle of attack, there is a monotonic iterative pattern, and for the bank angle, there is an overshoot pattern. All of the segments in the linear programming form are formally considered independent of each other, but the linear programming solutions are formed from control laws for which a high-frequency chattering is absent. In a sense, it is a bang–bang control.

Table 1 Performance index at iterations

Iteration number	Cross range, km
Initial guess	129.03
1	149.46
2	153.86
3	159.41
4	163.01
5	165.39
6	165.43

C. Comparison Example for Orbit Transfer Problem

As a comparison example with known solutions, we consider the maximum-radius orbit transfer described on the classic text of Bryson and Ho [1]. This problem is summarized in the text as follows: "Given a constant thrust rocket engine, $T = \text{const}$, operating for a given length of time, t_f, we wish to find the thrust-direction history, $\phi(t)$, to transfer a rocket vehicle from a given circular orbit to the largest possible circular orbit" (Fig. 13).

The equations of motion for the orbit transfer [1] are

$$\begin{cases} \dot r = u \\ \dot\theta = v/r \\ \dot u = \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T\sin(\phi)}{m_0 - \dot m t} \\ \dot v = -\frac{uv}{r} + \frac{T\cos(\phi)}{m_0 - \dot m t} \end{cases} \quad (39)$$

where u and v are the radial and tangential velocities, θ is the angular displacement, m_o is the initial mass, and $\dot m$ is the mass flow rate. The thrust angle is defined relative to the local horizontal direction at the spacecraft's position. The in-plane angle ϕ is the scalar control law for this problem.

In this example, the dimensionless time unit is $\sqrt{r_0^3/\mu}$, r_o is the astronomical unit, the gravitational constant is $\mu = 1$ (the sun's

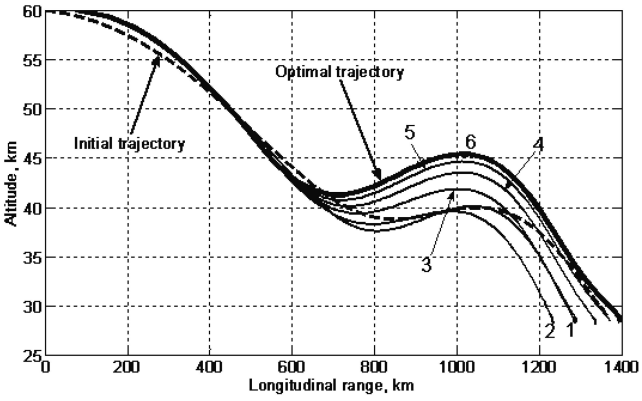

Fig. 8 Trajectory profile in vertical plane for each iteration.

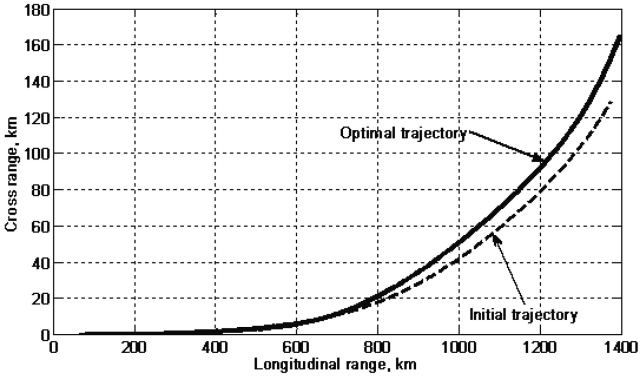

Fig. 9 Trajectory profile in horizontal plane.

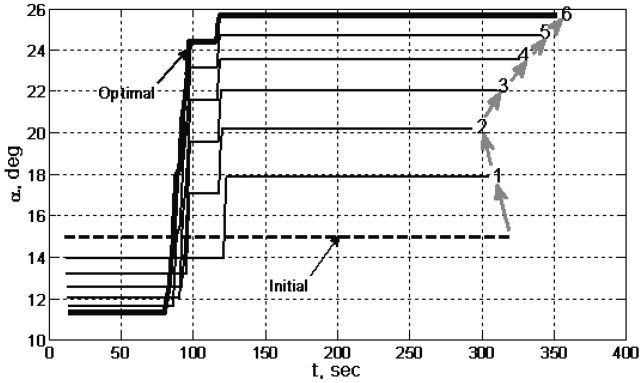

Fig. 10 Angle of attack history for each iteration.

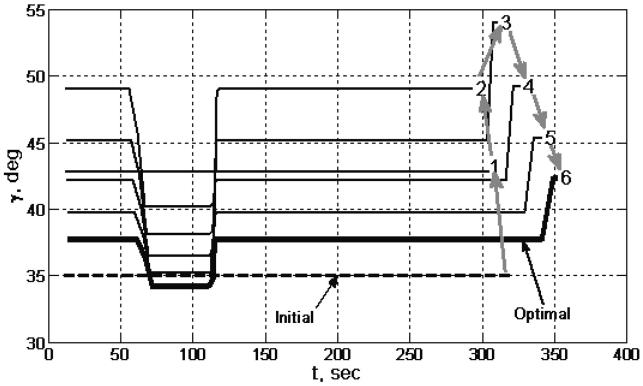

Fig. 11 Bank angle history for each iteration.

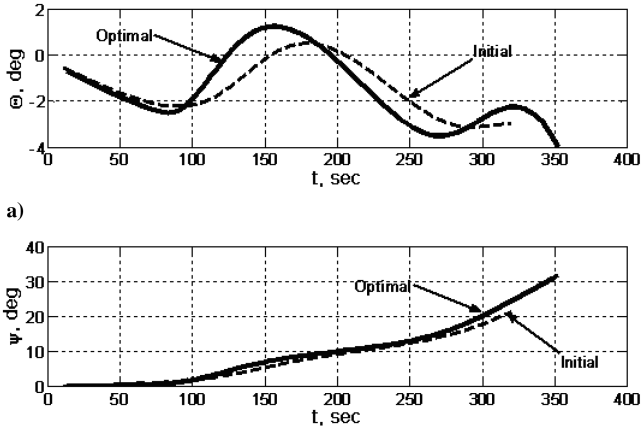

Fig. 12 Angles histories: a) flight-path angle and b) heading angle.

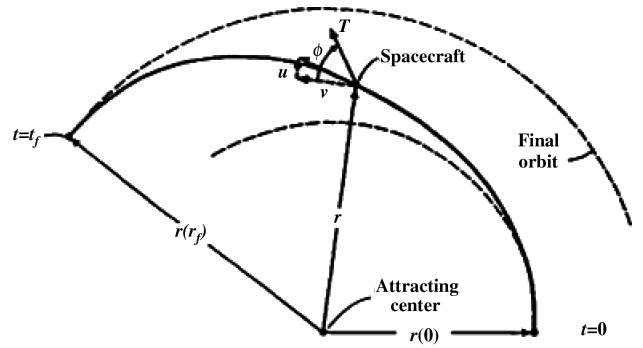

Fig. 13 Maximum-radius orbit transfer (reprinted with permission from [1]).

gravitational constant is $132,712,441,933\ \text{km}^3/\text{s}^2$, and the astronomical unit is 149,597,870.7 km), and the initial dimensionless mass is $m_0 = 1$.

The initial conditions correspond to the spacecraft's position and velocity while in the Earth's orbit:

$$r(0) = r_0 = 1.0, \qquad \theta(0) = 0.0, \qquad u(0) = 0.0, \qquad v(0) = 1.0 \tag{40}$$

and terminal boundary conditions are

$$\begin{cases} u(t_f) = 0 \\ v(t_f) = \sqrt{\mu / r(t_f)} \end{cases} \tag{41}$$

where $t_f = 193$ days is the specified final time.

Then fuel flow rate $\dot{m}$ is -0.07487, and the initial dimensionless acceleration is given by

$$\frac{T/m_0}{\mu/r_0^2} = 0.1405 \tag{42}$$

For this problem, we would like to maximize the radius of the final circular orbit. Therefore, the performance index is $J = -r(t_f)$. The problem can be rewritten in terms of nonsingular elements (eccentricity vector) [22]:

$$P_1 = e\sin(\omega_\pi); \qquad P_2 = e\cos(\omega_\pi) \tag{43}$$

where e is the orbit eccentricity and ω_π is the argument of periapsis. Then the terminal conditions [Eq. (41)] are

$$P_1(t_f) = 0; \qquad P_2(t_f) = 0 \tag{44}$$

and the performance index is

$$J = -a(t_f) \tag{45}$$

where a is the semimajor axis.

It should be noted that the problem is nonlinear and requires an iterative solution (i.e., a sequence of linear programming solutions). Suppose that there is a reference trajectory. The transfer problem is discretized into $n = 100$ segments, with the discrete set of pseudocontrols $\phi_i^{(j)}$ with a small angle of $\Delta\phi = 2\pi/k$ between them, where $(k = 360)$. Therefore, the number of decision variables is $(100 \times 360) = 36,000$. For an ith segment and jth pseudocontrol, the partial derivations [Eq. (14)] based on nonsingular form of the variational equations [22] can be expressed as

Table 2 Performance index and terminal parameters at iterations

Method	Iteration	r_f	θ_f	u_f	v_f	e_f
Sequence of linear programming solutions	Initial	1.000	——	0.000	1.000	0.00000
	1	2.064	144.4	0.083	0.768	0.15102
	2	1.506	145.0	0.012	0.788	0.02833
	3	1.537	145.5	0.023	0.793	0.02690
	4	1.474	145.0	−0.009	0.809	0.01768
	5	1.508	146.3	−0.008	0.801	0.01583
	Final (6)	1.511	148.6	0.007	0.808	0.00552
Indirect (two-point boundary-value problem [1])	Final	1.524	——	0.000	0.810	——
Direct (genetic algorithm [23])	Final	1.512	149.5	0.008	0.802	——

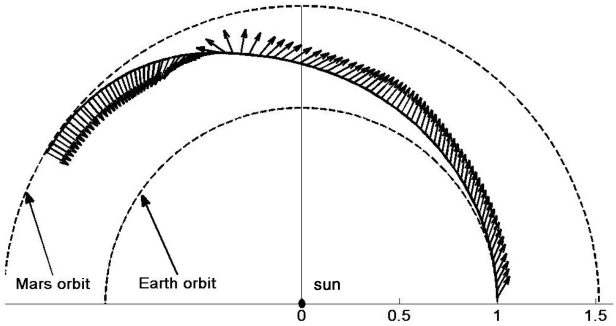

Fig. 14 Near-optimal Earth-to-Mars transfer.

$$\frac{\partial P_1}{\partial x_i^{(j)}} = \frac{r_i}{h_{ei}}\left\{-\frac{p_i}{r_i}\cos L_{\mathrm{ML}i}\cdot\sin[\phi_i^{(j)}] + \left[P_{1i} + \left(1+\frac{p_i}{r_i}\right)\sin L_{\mathrm{ML}i}\right]\right.$$
$$\left.\cdot\cos[\phi_i^{(j)}]\right\}\frac{a_0}{1-\Delta m\cdot i}\Delta t; \qquad \frac{\partial P_2}{\partial x_i^{(j)}} = \frac{r_i}{h_{ei}}\left\{\frac{p_i}{r_i}\sin L_{\mathrm{ML}i}\cdot\cos[\phi_i^{(j)}]\right.$$
$$\left.+\left[P_{2i} + \left(1+\frac{p_i}{r_i}\right)\cos L_{\mathrm{ML}i}\right]\cdot\sin[\phi_i^{(j)}]\right\}\frac{a_0}{1-\Delta m\cdot i}\Delta t \tag{46}$$

where p is the orbit parameter, h_e is the energy integral, L_{ML} is the mean longitude [22], Δm is the mass change for each segment, and $\Delta t = t_f/n$ is the segment duration. The values of $r_i, h_{ei}, p_i, P_{1i}, P_{2i}$, and $L_{\mathrm{ML}i}$ can be computed along the reference trajectory. The weight coefficients for the performance index [Eq. (21)] are

$$q_i^{(j)} = \frac{2a_i^2}{h_{ei}}\left\{(P_{2i}\sin L_{\mathrm{ML}i} - P_{1i}\cos L_{\mathrm{ML}i})\sin[\phi_i^{(j)}] + \frac{p_i}{r_i}\cos[\phi_i^{(j)}]\right\}$$
$$\cdot\frac{a_0}{1-\Delta m\cdot i}\Delta t \tag{47}$$

Then the matrix $\mathbf{A_e}$ for the linear matrix (15) can be presented as

$$\mathbf{A_e} = \begin{bmatrix} \underbrace{111\cdots1}_{k} & & & & \\ & \underbrace{111\cdots1}_{k} & & & \\ & & & \cdots & \\ & & & & \underbrace{111\cdots1}_{k} \\ \frac{\partial P_1}{\partial x_1^{(1)}} & \frac{\partial P_1}{\partial x_1^{(2)}} & \cdots & \cdots & \frac{\partial P_1}{\partial x_n^{(k)}} \\ \frac{\partial P_2}{\partial x_1^{(1)}} & \frac{\partial P_2}{\partial x_1^{(2)}} & \cdots & \cdots & \frac{\partial P_2}{\partial x_n^{(k)}} \end{bmatrix} \tag{48}$$

and the $(n+2)$-dimensional vector is

$$\mathbf{P}_d^T = [1, \quad 1, \quad 1\cdots1, \quad 1, \quad \delta P_1 \quad \delta P_2] \tag{49}$$

where δP_1 and δP_2 are target values.

As the reference trajectory for computation of the partial derivatives, we use the initial circular orbit of the Earth. Based on the linear programming solution and postprocessing, we compute a new reference trajectory using the variational equations for nonsingular elements [22]. The partial derivatives [Eq. (46)] and weight coefficients [Eq. (47)] are refined for the second iteration, etc., to obtain an acceptable solution. For the first three iterations, the target values are $\delta P_1 = \delta P_2 = 0$ and further

$$\delta P_1 = -P_1^*(t_f); \qquad \delta P_2 = -P_2^*(t_f) \tag{50}$$

where $P_1^*(t_f)$ and $P_2^*(t_f)$ are the terminal values from the previous iteration.

The optimization results for six iterations are given in Table 2. The last column of the table is the final eccentricity $e_f = \sqrt{P_1^2 + P_2^2}$. The full solution requires ~1.5 min of computation time using a MATLAB implementation on a Pentium IV processor (3.0 GHz).

The optimal thrust directions for each segment and orbit transfer trajectory are shown in Fig. 14.

For comparison, examples of results using two other methods are also presented in the last two rows of Table 2. The first is the solution using an indirect method based on the classical two-point boundary-value problem [1]. The second is use of a direct method based on genetic algorithms [23]. It is shown that the final results are almost coincident.

V. Conclusions

In this paper, a new concept of pseudocontrol sets to solve optimal control problems is proposed. The approach is based on discretizing the system motion on small segments, and the key idea is an application of the control space discrete approximation by a set of pseudocontrol vectors for each segment. Another feature of the approach is a significant increase in the number of the decision variables by introducing artificial variables or pseudovariables, but at the same time, it permits a transformation of the optimal control problem into a classical linear programming form. The proposed technique provides flexible possibilities for optimal control problem with various constraints including interior-point constraints. Use of this method for a wider range of nonlinear problems should be investigated.

Appendix A: Postprocessing Algorithm for Scalar Control Function

Suppose that the same pseudocontrol sets ($u^{(j)}$, $j = \overline{1,k}$) are for each segment. As input data, a $n \times k$ vector of linear programming solution $\mathbf{X}$ is used. Additional variables are listed in Table A1.

A postprocessing algorithm is found in Fig. A1.

Table A1 Variable definitions

Variable	Definition
NumVar	Number of current decision variable
NumValSeg	Number of the nonzero variables for current segment
TolSol	Tolerance for choice of the nonzero variables ($\sim 10^{-3}$–10^{-5})
mag$X$	Summary magnitude of the nonzero decision variables at arbitrary segment
jPrev	Index of previous nonzero variable
Cont(i)	Control at each segment, $j = \overline{1,n}$

```
INPUT: X,U,n,k
For i=1,n                                   % Cycle for  segments
    NumValSeg=0
    For j=1,k                               % Cycle for  pseudo-controls
      NumVar=(i-1)·k+j                      % Current number of decision variable
      If X(NumVar)>TolSol
          % Variable is non-zero
          NumValSeg=NumValSeg+1
          Cont(i)=U(j)
          modX=X(NumVar)
          If NumValSeg=2
                If jPrev+1=j
                      % Two adjacent non-zero variables
                      Cont(i)=U(j)·X(NumVar)+U(jPrev)·modX
                       modX=ModX+X(NumVar)
                Else
                      Message(Two non-adjacent, non-zero variables <j> and
                                  <jPrev> at <i> segment)
                      STOP
                End
          End
          If NumValSeg>2
                Message(Invalid number of pseudo-control at ⟨i⟩segment)
                STOP
          End
          jPrev=j
    End
  End                                       % Cycle for  pseudo-controls
End                                         % Cycle for  segments
OUTPUT: Cont
```

Fig. A1 Postprocessing algorithm.

Acknowledgment

The author thanks Robert Melton for constructive comments and help in preparation of the paper.

References

[1] Bryson, A. E., Jr., and Ho, Y.-C., *Applied Optimal Control*, Hemisphere, New York, 1975, Chaps. 2.5, and 7.3.

[2] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., *The Mathematical Theory of Optimal Processes*, Wiley, New York, 1962, Chap. 2.

[3] Betts, J., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi:10.2514/2.4231

[4] Dantzig, G., *Linear Programming and Extensions*, Princeton Univ. Press, Princeton, NJ, 1963, Chaps. 2–3.

[5] Dantzig, G., "Linear Control Processes and Mathematical Programming," *SIAM Journal on Control and Optimization*, Vol. 4, No. 1, 1966, pp. 56–60. doi:10.1137/0304006

[6] Ulybyshev, Y. P., and Sokolov, A. V., "Many-Revolution, Low-Thrust Maneuvers in Vicinity of Geostationary Orbit," *Journal of Computer and Systems Sciences International*, Vol. 38, No. 2, 1999, pp. 255–261.

[7] Bodson, M., "Evaluation of Optimization Methods for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711. doi:10.2514/2.4937

[8] Petersen, J., and Bodson, M., "Interior-Point Algorithms for Control Allocation," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp. 471–480. doi:10.2514/1.5937

[9] Kim, J-J., and Singh, T., "Desensitized Control of Vibratory System with Friction: Linear Programming Approach," *Optimal Control Applications and Methods*, Vol. 25, No. 4, 2004, pp. 165–180. doi:10.1002/oca.743

[10] Driessen, B. J., "On–Off Minimum-Time Control with Limited Fuel Usage: Near Global Optima via Linear Programming," *Optimal Control Applications and Methods*, Vol. 27, No. 3, 2006, pp. 161–168. doi:10.1002/oca.776

[11] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., "Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Linear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764. doi:10.2514/2.4943

[12] Khachian, L. G., "A Polynomial Algorithm for Linear Programming," *Comptes Rendus de l'Academie des Sciences de l'URSS / Doklady Akademii Nauk SSSR*, Vol. 244, No. , 1979, pp. 1093–1096; also *Soviet Mathematics Doklady*, Vol. 20, No. 2, 1979, pp. 191–194 (in English).

[13] Karmarkar, N., "A New Polynomial-Time Algorithm for Linear Programming," *Combinatorica : an International Journal of the Janos Bolyai Mathematical Society*, Vol. 4, No. 4, 1984, p. 373. doi:10.1007/BF02579150

[14] Wright, M. H., "The Interior-Point Revolution in Optimization: History, Recent Developments, and Lasting Consequences," *Bulletin of the American Mathematical Society*, Vol. 42, No. 1, 2004, pp. 39–56. doi:10.1090/S0273-0979-04-01040-7

[15] Rao, S. S., and Mulkay, E. L., "Engineering Design Optimization Using Interior-Point Algorithms," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2127–2132. doi:10.2514/2.875

[16] Ulybyshev, Y., "Continuous Thrust Orbit Transfer Optimization Using Large-Scale Linear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 427–436. doi:10.2514/1.22642

[17] Ulybyshev, Y., "Optimization of Multi-Mode Rendezvous Trajectories with Constraints," *Cosmic Research (Translation of Kosmicheskie Issledovaniya)*, Vol. 46, No. 2, 2008, pp. 133–147. doi:10.1134/S0010952508020056

[18] Ulybyshev, Y., "Spacecraft Trajectory Optimization Based on Discrete Sets of Pseudo-Impulses," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 4, 2009, pp. 1209–1217. doi:10.2514/1.41201; see also AIAA Paper 2008-6276, 2008.

[19] MATLAB® Users Guide, MathWorks, Natick, MA, 2003, Chap. 16.

[20] Betts, J., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2001, Chaps. 1.11.1, 1.4, 4.1.4.

[21] Bharadwaj, S., Rao, A. V., and Mease, K. D., "Entry Trajectory Tracking Law via Feedback Linearization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 726–732. doi:10.2514/2.4318

[22] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, Reston, VA, 1999, Chap. 10.4.

[23] Rauwolf, G. A., and Coverstone-Carrol, V. K., "Near-Optimal Lowthrust Orbit Transfers Generated by a Genetic Algorithm," *Journal of Spacecraft and Rockets*, Vol. 33, No. 6, 1996, pp. 859–862. doi:10.2514/3.26850