JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 21, No. 2, March–April 1998

Survey of Numerical Methods for Trajectory Optimization

John T. Betts
Boeing Information and Support Services, Seattle, Washington 98124-2207

I. Introduction

IT is not surprising that the development of numerical methods for trajectory optimization have closely paralleled the exploration of space and the development of the digital computer. Space exploration provided the impetus by presenting scientists and engineers with challenging technical problems. The digital computer provided the tool for solving these new problems. The goal of this paper is to review the state of the art in the field loosely referred to as trajectory optimization.

Presenting a survey of a field as diverse as trajectory optimization is a daunting task. Perhaps the most difficult issue is restricting the scope of the survey to permit a meaningful discussion within a limited amount of space. To achieve this goal, I made a conscious decision to focus on the two types of methods most widely used today, namely, direct and indirect. I begin the discussion with a brief review of the underlying mathematics in both direct and indirect methods. I then discuss the complications that occur when path and boundary constraints are imposed on the problem description. Finally, I describe unresolved issues that are the subject of ongoing research.

A number of recurrent themes appear throughout the paper. First, the aforementioned direct vs indirect is introduced as a means of categorizing an approach. Unfortunately, not every technique falls neatly into one category or another. I will attempt to describe the benefits and deficiencies in both approaches and then suggest that the techniques may ultimately merge. Second, I shall attempt to discriminate between method vs implementation. A numerical method is usually described by mathematical equations and/or algorithmic logic. Computational results are achieved by implementing the algorithm as software, e.g., Fortran code. A second level of implementation may involve translating a (preflight) scientific software implementation into an (onboard) hardware implementation. In general, method and implementation are not the same, and I shall try to emphasize that fact. Third, I shall focus the discussion on algorithms instead of physical models. The definition of a trajectory problem necessarily entails a definition of the dynamic environment such as gravitational, propulsion, and aerodynamic forces. Thus it is common to use the same algorithm, with different physical models, to solve different problems. Conversely, different algorithms may be applied to the same physical models (with entirely different results). Finally, I shall attempt to focus on general rather than special purpose methods. A great deal of research has been directed toward solving specific problems. Carefully specialized techniques can either be very effective or very ad hoc. Unfortunately, what works well for a launch vehicle guidance problem may be totally inappropriate for a low-thrust orbit transfer.

II. Trajectory Optimization Problem

Let us begin the discussion by defining the problem in a fairly general way. A trajectory optimization or optimal control problem can be formulated as a collection of N phases. In general, the independent variable t for phase k is defined in the region $t_0^{(k)} \le t \le t_f^{(k)}$. For many applications the independent variable t is time, and the phases are sequential, that is, $t_0^{(k+1)} = t_f^{(k)}$; however, neither of those assumptions is required. Within phase k the dynamics of the system are described by a set of dynamic variables

$$z = \begin{bmatrix} \boldsymbol{y}^{(k)}(t) \\ \boldsymbol{u}^{(k)}(t) \end{bmatrix} \tag{1}$$

made up of the $n_y^{(k)}$ state variables and the $n_u^{(k)}$ control variables, respectively. In addition, the dynamics may incorporate the $n_p^{(k)}$ parameters $\boldsymbol{p}^{(k)}$, which are not dependent on t. For clarity I drop the phase-dependent notation from the remaining discussion in this section. However, it is important to remember that many complex problem descriptions require different dynamics and/or constraints, and a phase-dependent formulation accommodates this requirement.

Typically the dynamics of the system are defined by a set of ordinary differential equations written in explicit form, which are referred to as the state or system equations,

$$\dot{\boldsymbol{y}} = \boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{p}, t] \tag{2}$$

where $\boldsymbol{y}$ is the n_y dimension state vector. Initial conditions at time t_0 are defined by

$$\psi_{0l} \le \psi[\boldsymbol{y}(t_0), \boldsymbol{u}(t_0), \boldsymbol{p}, t_0] \le \psi_{0u} \tag{3}$$

where $\psi[\boldsymbol{y}(t_0), \boldsymbol{u}(t_0), \boldsymbol{p}, t_0] \equiv \psi_0$, and terminal conditions at the final time t_f are defined by

$$\psi_{fl} \le \psi[\boldsymbol{y}(t_f), \boldsymbol{u}(t_f), \boldsymbol{p}, t_f] \le \psi_{fu} \tag{4}$$

where $\psi[\boldsymbol{y}(t_f), \boldsymbol{u}(t_f), \boldsymbol{p}, t_f] \equiv \psi_f$. In addition, the solution must satisfy algebraic path constraints of the form

$$\boldsymbol{g}_l \le \boldsymbol{g}[\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{p}, t] \le \boldsymbol{g}_u \tag{5}$$

where $\boldsymbol{g}$ is a vector of size n_g, as well as simple bounds on the state variables

$$\boldsymbol{y}_l \le \boldsymbol{y}(t) \le \boldsymbol{y}_u \tag{6}$$

John T. Betts received a B.A. degree from Grinnell College, Grinnell, Iowa, in 1965 with a major in physics and a minor in mathematics. In 1967 he received an M.S. in astronautics from Purdue University, West Lafayette, Indiana, with a major in orbit mechanics, and in 1970 he received a Ph.D. from Purdue, specializing in optimal control theory. He joined The Aerospace Corporation in 1970 as a Member of the Technical Staff and from 1977–1987 was manager of the Optimization Techniques Section of the Performance Analysis Department. He joined The Boeing Company, serving as manager of the Operations Research Group of Boeing Computer Services from 1987–1989. In his current position as Senior Principal Scientist in the Applied Research and Technology Division, he provides technical support to all areas of The Boeing Company. Dr. Betts is a Member of AIAA and the Society for Industrial and Applied Mathematics with active research in nonlinear programming and optimal control theory. E-mail: John.T.Betts@boeing.com.

Received March 29, 1997; revision received Oct. 7, 1997; accepted for publication Oct. 20, 1997.

and control variables

$$\boldsymbol{u}_l \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_u \tag{7}$$

Note that an equality constraint can be imposed if the upper and lower bounds are equal, e.g., $(g_l)_k = (g_u)_k$ for some k.

Finally, it may be convenient to evaluate expressions of the form

$$\int_{t_0}^{t_f} \boldsymbol{q}[\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{p}, t]\,\mathrm{d}t \tag{8}$$

which involve the quadrature functions $\boldsymbol{q}$. Collectively we refer to those functions evaluated during the phase, namely,

$$\boldsymbol{F}(t) = \begin{bmatrix} \boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{p}, t] \\ \boldsymbol{g}[\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{p}, t] \\ \boldsymbol{q}[\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{p}, t] \end{bmatrix} \tag{9}$$

as the vector of continuous functions. Similarly, functions evaluated at a specific point, such as the boundary conditions $\boldsymbol{\psi}[\boldsymbol{y}(t_0), \boldsymbol{u}(t_0), t_0]$ and $\boldsymbol{\psi}[\boldsymbol{y}(t_f), \boldsymbol{u}(t_f), t_f]$, are referred to as point functions.

The basic optimal control problem is to determine the $n_u^{(k)}$-dimensional control vectors $\boldsymbol{u}^{(k)}(t)$ and parameters $\boldsymbol{p}^{(k)}$ to minimize the performance index

$$J = \phi\Big[\boldsymbol{y}\big(t_0^{(1)}\big), t_0^{(1)}, \boldsymbol{y}\big(t_f^{(1)}\big), \boldsymbol{p}^{(1)}, t_f^{(1)}, \ldots, \boldsymbol{y}\big(t_0^{(N)}\big), t_0^{(N)}, \boldsymbol{y}\big(t_f^{(N)}\big), \boldsymbol{p}^{(N)}, t_f^{(N)}\Big] \tag{10}$$

Notice that the objective function may depend on quantities computed in each of the N phases.

This formulation raises a number of points that deserve further explanation. The concept of a phase, also referred to as an arc by some authors, partitions the time domain. In this formalism the differential equations cannot change within a phase but may change from one phase to another. An obvious reason to introduce a phase is to accommodate changes in the dynamics, for example, when simulating a multistage rocket. The boundary of a phase is often called an event or junction point. A boundary condition that uniquely defines the end of a phase is sometimes called an event criterion (and a well-posed problem can have only one criterion at each event). Normally, the simulation of a complicated trajectory may link phases together by forcing the states to be continuous, e.g., $\boldsymbol{y}[t_f^{(1)}] = \boldsymbol{y}[t_0^{(2)}]$. However, for multipath or branch trajectories this may not be the case.[1] The differential equations (2) have been written as an explicit system of first-order equations, i.e., with the first derivative appearing explicitly on the left-hand side, which is the standard convention for aerospace applications. Although this simplifies the presentation, it may not be either necessary or desirable; e.g., Newton's law $\boldsymbol{F} = m\boldsymbol{a}$ is not stated as an explicit first-order system! The objective function (10) has been written in terms of quantities evaluated at the ends of the phases, and this is referred to as the Mayer form.[2] If the objective function only involves an integral (8), it is referred to as a problem of Lagrange, and when both terms are present, it is called a problem of Bolza. It is well known that the Mayer form can be obtained from either the Lagrange or Bolza form by introducing an additional state variable. However, again this may have undesirable numerical consequences.

III. Nonlinear Programming

A. Newton's Method

Essentially all numerical methods for solving the trajectory optimization problem incorporate some type of iteration with a finite set of unknowns. In fact, progress in optimal control solution methods closely parallels the progress made in the underlying nonlinear programming (NLP) methods. Space limitations preclude an in-depth presentation of constrained optimization methods. However, it is important to review some of the fundamental concepts. For more complete information the reader is encouraged to refer to the books by Fletcher,[3] Gill et al.,[4] and Dennis and Schnabel.[5] The fundamental approach to most iterative schemes was suggested over 300 years ago by Newton. Suppose we are trying to solve the nonlinear algebraic equations $\boldsymbol{a}(\boldsymbol{x}) = 0$ for the root $\boldsymbol{x}^*$. Beginning with an estimate $\boldsymbol{x}$ we can construct a new estimate $\bar{\boldsymbol{x}}$ according to

$$\bar{\boldsymbol{x}} = \boldsymbol{x} + \alpha\boldsymbol{p} \tag{11}$$

where the search direction $\boldsymbol{p}$ is computed by solving the linear system

$$\boldsymbol{A}(x)\boldsymbol{p} = -\boldsymbol{a}(x) \tag{12}$$

The $n \times n$ matrix $\boldsymbol{A}$ is defined by

$$\boldsymbol{A} = \begin{bmatrix} \dfrac{\partial a_1}{\partial x_1} & \dfrac{\partial a_1}{\partial x_2} & \cdots & \dfrac{\partial a_1}{\partial x_n} \\ \dfrac{\partial a_2}{\partial x_1} & \dfrac{\partial a_2}{\partial x_2} & \cdots & \dfrac{\partial a_2}{\partial x_n} \\ \vdots & & \ddots & \\ \dfrac{\partial a_n}{\partial x_1} & \dfrac{\partial a_n}{\partial x_2} & \cdots & \dfrac{\partial a_n}{\partial x_n} \end{bmatrix} \tag{13}$$

When the scalar step length α is equal to 1, the iteration scheme is equivalent to replacing the nonlinear equation by a linear approximation constructed about the point $\boldsymbol{x}$. We expect the method to converge provided the initial guess is close to the root $\boldsymbol{x}^*$. Of course, this simple scheme is not without pitfalls. First, to compute the search direction, the matrix $\boldsymbol{A}$ must be nonsingular (invertible), and for arbitrary nonlinear functions $\boldsymbol{a}(x)$ this may not be true. Second, when the initial guess is not close to the root, the iteration may diverge. One common way to stabilize the iteration is to reduce the length of the step by choosing α such that

$$\|\boldsymbol{a}(\bar{\boldsymbol{x}})\| \leq \|\boldsymbol{a}(\boldsymbol{x})\| \tag{14}$$

The procedure for adjusting the step length is called a line search, and the function used to measure progress (in this case $\|\boldsymbol{a}\|$) is called a merit function. In spite of the need for caution, Newton's method enjoys broad applicability, possibly because, when it works, the iterates exhibit quadratic convergence. Loosely speaking, this property means that the number of significant figures in $\boldsymbol{x}$ (as an estimate for $\boldsymbol{x}^*$) doubles from one iteration to the next.

B. Unconstrained Optimization

Let us now consider an unconstrained optimization problem. Suppose that we must choose the n variables $\boldsymbol{x}$ to minimize the scalar objective function $F(\boldsymbol{x})$. Necessary conditions for $\boldsymbol{x}^*$ to be a stationary point are

$$\boldsymbol{g}(\boldsymbol{x}^*) = \nabla_x F = \begin{bmatrix} \dfrac{\partial F}{\partial x_1} \\ \dfrac{\partial F}{\partial x_2} \\ \vdots \\ \dfrac{\partial F}{\partial x_n} \end{bmatrix} = 0 \tag{15}$$

Now if Newton's method is used to find a point where the gradient (15) is zero, we must compute the search direction using

$$\boldsymbol{H}(x)\boldsymbol{p} = -\boldsymbol{g}(x) \tag{16}$$

where the Hessian matrix $\boldsymbol{H}$ is the symmetric matrix of second derivatives of the objective function. Just as before, there are pitfalls in using this method to construct an estimate of the solution. First, we note that the condition $\boldsymbol{g} = 0$ is necessary but not sufficient. Thus a point with zero gradient can be either a maximum or a minimum. At a minimum point the Hessian matrix is positive definite, but this may not be true when $\boldsymbol{H}$ is evaluated at some point far from the solution. In fact, it is quite possible that the direction $\boldsymbol{p}$ computed by solving Eq. (16) will point uphill rather than downhill. Second, there is some ambiguity in the choice of a merit function if a line search is used to stabilize the method. Certainly we would hope to reduce the objective function, that is, $F(\bar{x}) \le F(x)$. However, this may not produce a decrease in the gradient

$$\|g(\bar{x})\| \le \|g(x)\| \tag{17}$$

In fact, what have been described are two issues that distinguish a direct and an indirect method for finding a minimum. For an indirect method, a logical choice for the merit function is $\|g(x)\|$. In contrast, for a direct method, one probably would insist that the objective function is reduced at each iteration, and to achieve this it may be necessary to modify the calculation of the search direction to ensure that it is downhill. A consequence of this is that the region of convergence for an indirect method may be considerably smaller than the region of convergence for a direct method. Stated differently, an indirect method may require a better initial guess than required by a direct method. Second, to solve the equations $g = 0$, it is necessary to compute the expressions $g(x)$! Typically this implies that analytic expressions for the gradient are necessary when using an indirect method. In contrast, finite difference approximations to the gradient are often used in a direct method.

C. Equality Constraints

Suppose that we must choose the n variables x to minimize the scalar objective function $F(x)$ and satisfy the m equality constraints

$$c(x) = 0 \tag{18}$$

where $m \le n$. We introduce the Lagrangian

$$L(x, \lambda) = F(x) - \lambda^\top c(x) \tag{19}$$

which is a scalar function of the n variables x and the m Lagrange multipliers λ. Necessary conditions for the point (x^*, λ^*) to be a constrained optimum require finding a stationary point of the Lagrangian that is defined by

$$\nabla_x L(x, \lambda) = g(x) - G^\top(x)\lambda = 0 \tag{20}$$

and

$$\nabla_\lambda L(x, \lambda) = -c(x) = 0 \tag{21}$$

By analogy with the development in the preceding sections, we can use Newton's method to find the $(n + m)$ variables (x, λ) that satisfy the conditions (20) and (21). Proceeding formally to construct the linear system equivalent to Eq. (12), one obtains

$$\begin{bmatrix} H_L & G^\top \\ G & 0 \end{bmatrix} \begin{bmatrix} p \\ -\bar{\lambda} \end{bmatrix} = \begin{bmatrix} -g \\ -c \end{bmatrix} \tag{22}$$

This system requires the Hessian of the Lagrangian

$$H_L = \nabla_x^2 F - \sum_{i=1}^{m} \lambda_i \nabla_x^2 c_i \tag{23}$$

The linear system (22) is referred to as the Kuhn–Tucker (KT) or Karush–Kuhn–Tucker system. It is important to observe that an equivalent way of defining the search direction p is to minimize the quadratic

$$\tfrac{1}{2} p^\top H_L p + g^\top p \tag{24}$$

subject to the linear constraints

$$Gp = -c \tag{25}$$

This is referred to as a quadratic programming (QP) subproblem. Just as in the unconstrained and root solving applications discussed earlier, when Newton's method is applied to equality constrained problems, it may be necessary to modify either the magnitude of the step via a line search or the direction itself using a trust region approach. However, regardless of the type of stabilization invoked at points far from the solution, near the answer all methods try to mimic the behavior of Newton's method.

D. Inequality Constraints

An important generalization of the preceding problem occurs when inequality constraints are imposed. Suppose that we must choose the n variables x to minimize the scalar objective function $F(x)$ and satisfy the m inequality constraints

$$c(x) \ge 0 \tag{26}$$

In contrast to the equally constrained case, now m may be greater than n. However, at the optimal point x^*, the constraints will fall into one of two classes. Constraints that are strictly satisfied, i.e., $c_i(x^*) > 0$, are called inactive. The remaining active constraints are on their bounds, i.e., $c_i(x^*) = 0$. If the active set of constraints is known, then one can simply ignore the remaining constraints and treat the problem using methods for an equality constrained problem. However, algorithms to efficiently determine the active set of constraints are nontrivial because they require repeated solution of the KT system (22) as constraints are added and deleted. In spite of the complications, methods for nonlinear programming based on the solution of a series of quadratic programming subproblems are widely used. A popular implementation of the so-called successive or sequential quadratic programming (SQP) approach is found in the software NPSOL.[6,7]

In summary, the NLP problem requires finding the n vector x to minimize

$$F(x) \tag{27}$$

subject to the m constraints

$$c_L \le c(x) \le c_U \tag{28}$$

and bounds

$$x_L \le x \le x_U \tag{29}$$

In this formulation, equality constraints can be imposed by setting $c_L = c_U$.

E. Historical Perspective

Progress in the development of NLP algorithms has been closely tied to the advent of the digital computer. Because NLP software is a primary piece of the trajectory optimization tool kit, it has been a pacing item in the development of sophisticated trajectory optimization software. In the early 1960s most implementations were based on a simple Newton method (11) and (12) with optimization done parametrically, i.e., by hand. The size of a typical application was $n = m \approx 10$. In the 1970s quasi-Newton approximations[3,5] became prevalent. One popular approach for dealing with constraints was to apply an unconstrained minimization algorithm to a modified form of the objective, e.g., minimize $J(x, \rho) = F(x) + \frac{1}{2}\rho c(x)^\top c(x)$, where ρ is large. Although those techniques have generally been superseded for general optimization, curiously enough they are fundamental to the definition of the merit functions used to stabilize state-of-the-art algorithms. A second popular approach for constrained problems, referred to as the reduced gradient approach, identifies a basic set of variables that are used to eliminate the active constraints, permitting choice of the nonbasic variables using an unconstrained technique. Careful implementations of this method[8–10] can be quite effective, especially when the constraints are nearly linear and the number of inequalities is small. Most applications in the 1970s and early 1980s were of moderate size, i.e., $n = m < 100$. Current applications have incorporated advances in numerical linear algebra that exploit matrix sparsity, thereby permitting applications with $n, m \approx 100{,}000$ (Refs. 11–20).

IV. Optimal Control

A. Dynamic Constraints

The optimal control problem may be interpreted as an extension of the nonlinear programming problem to an infinite number of variables. For fundamental background in the associated calculus of variations the reader should refer to Ref. 21. First let us consider a simple problem with a single phase and no path constraints.

Specifically, suppose we must choose the control functions $\boldsymbol{u}(t)$ to minimize

$$J = \phi[\boldsymbol{y}(t_f), t_f] \tag{30}$$

subject to the state equations

$$\dot{\boldsymbol{y}} = \boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t)] \tag{31}$$

and the boundary conditions

$$\boldsymbol{\psi}[\boldsymbol{y}(t_f), \boldsymbol{u}(t_f), t_f] = 0 \tag{32}$$

where the initial conditions $\boldsymbol{y}(t_0) = \boldsymbol{y}_0$ are given at the fixed initial time t_0, and the final time t_f is free. Note that this is a very simplified version of the problem (2–10), and we have purposely chosen a problem with only equality constraints. However, in contrast to the previous discussion, we now have a continuous equality constraint (31) as well as a discrete equality (32). In a manner analogous to the definition of the Lagrangian function (19), we form an augmented performance index

$$\hat{J} = [\phi + \boldsymbol{\nu}^\top \boldsymbol{\psi}]_{t_f} + \int_{t_0}^{t_f} \boldsymbol{\lambda}^\top(t)\{\boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t)] - \dot{\boldsymbol{y}}\}\,\mathrm{d}t \tag{33}$$

Notice that, in addition to the Lagrange multipliers $\boldsymbol{\nu}$ for the discrete constraints, we have multipliers $\boldsymbol{\lambda}(t)$ referred to as adjoint or costate variables for the continuous (differential equation) constraints. In the finite dimensional case, the necessary conditions for a constrained optimum (20) and (21) were obtained by setting the first derivatives of the Lagrangian to zero. The analogous operation is to set the first variation $\delta\hat{J} = 0$. It is convenient to define the Hamiltonian

$$H = \boldsymbol{\lambda}^\top(t)\boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t)] \tag{34}$$

and the auxiliary function

$$\boldsymbol{\Phi} = \phi + \boldsymbol{\nu}^\top \boldsymbol{\psi} \tag{35}$$

The necessary conditions referred to as the Euler–Lagrange equations that result from setting the first variation to zero in addition to Eqs. (31) and (32) are

$$\dot{\boldsymbol{\lambda}} = -\boldsymbol{H}_y^\top \tag{36}$$

called the adjoint equations,

$$0 = \boldsymbol{H}_u^\top \tag{37}$$

called the control equations, and

$$\boldsymbol{\lambda}(t_f) = \boldsymbol{\Phi}_y^\top\big|_{t=t_f} \tag{38}$$

$$0 = (\boldsymbol{\Phi}_t + H)|_{t=t_f} \tag{39}$$

$$0 = \boldsymbol{\lambda}(t_0) \tag{40}$$

called the transversality conditions. The partial derivatives $\boldsymbol{H}_y$, $\boldsymbol{H}_u$, and $\boldsymbol{\Phi}_y$ are considered row vectors, i.e., $\boldsymbol{H}_y \doteq (\partial H/\partial y_1, \ldots, \partial H/\partial y_n)$, in these expressions. The control equations (37) are an application of the Pontryagin maximum principle.[22] A more general expression is

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}\in U} \boldsymbol{H} \tag{41}$$

where U defines the domain of feasible controls. Note that Eq. (41) is really a minimum principle to be consistent with the algebraic sign conventions used elsewhere. The maximum principle states that the control variable must be chosen to optimize the Hamiltonian (at every instant in time), subject to limitations on the control imposed by state and control path constraints. In essence, the maximum principle is a constrained optimization problem in the variables $\boldsymbol{u}(t)$ at all values of t. The complete set of necessary conditions consists of a differential-algebraic (DAE) system (31), (36), and (37) with boundary conditions at both t_0 and t_f in Eqs. (38), (39), and (32). This is often referred to as a two-point boundary value problem. A more extensive presentation of this material can be found in Ref. 23.

B. Algebraic Equality Constraints

Generalizing the problem in the preceding section, let us assume that we impose algebraic path constraints of the form

$$0 = \boldsymbol{g}[\boldsymbol{y}(t), \boldsymbol{u}(t), t] \tag{42}$$

in addition to the other conditions (31) and (32). Using notation similar to the preceding section, let us define the matrix

$$\boldsymbol{g}_u = \begin{bmatrix} \dfrac{\partial g_1}{\partial u_1} & \dfrac{\partial g_1}{\partial u_2} & \cdots & \dfrac{\partial g_1}{\partial u_n} \\ \dfrac{\partial g_2}{\partial u_1} & \dfrac{\partial g_2}{\partial u_2} & \cdots & \dfrac{\partial g_2}{\partial u_n} \\ \vdots & & \ddots & \\ \dfrac{\partial g_n}{\partial u_1} & \dfrac{\partial g_n}{\partial u_2} & \cdots & \dfrac{\partial g_n}{\partial u_n} \end{bmatrix} \tag{43}$$

Two possibilities exist. If the matrix $\boldsymbol{g}_u$ is full rank, then the system of differential and algebraic equations (31) and (42) is referred to as a DAE of index 1, and Eq. (42) is termed a control variable equality constraint. For this case the Hamiltonian (34) is replaced by

$$H = \boldsymbol{\lambda}^\top \boldsymbol{f} + \boldsymbol{\mu}^\top \boldsymbol{g} \tag{44}$$

which will result in modification to both the adjoint equations (36) and the control equations (37).

The second possibility is that the matrix $\boldsymbol{g}_u$ is rank deficient. In this case we can differentiate Eq. (42) with respect to t, yielding

$$0 = \boldsymbol{g}_y\dot{\boldsymbol{y}} + \boldsymbol{g}_u\dot{\boldsymbol{u}} + \boldsymbol{g}_t \tag{45}$$

$$= \boldsymbol{g}_y\boldsymbol{f}[\boldsymbol{y}, \boldsymbol{u}] + \boldsymbol{g}_u\dot{\boldsymbol{u}} + \boldsymbol{g}_t \tag{46}$$

$$\doteq \boldsymbol{g}'[\boldsymbol{y}(t), \boldsymbol{u}(t), t] \tag{47}$$

where the second step follows by substituting (31) and changing the definition of $\boldsymbol{y}$ and $\boldsymbol{u}$. The result is a new path constraint function $\boldsymbol{g}'$ that is mathematically equivalent provided that the original constraint is imposed at some point on the path, e.g., $0 = \boldsymbol{g}[\boldsymbol{y}(t_0), \boldsymbol{u}(t_0), t_0]$. For this new path function, again, the matrix $\boldsymbol{g}'_u$ may be full rank or rank deficient. If the matrix is full rank, the original DAE system is said to have index 2, and this is referred to as a state variable constraint of order 1. In the rank deficient case we may redefine the Hamiltonian using $\boldsymbol{g}'$ in place of $\boldsymbol{g}$. Of course, if the matrix $\boldsymbol{g}'_u$ is rank deficient, the process must be repeated. This is referred to as index reduction in the DAE literature.[24,25] Note that index reduction may be difficult to perform, and imposition of a high-index path constraint may be prone to numerical error.

C. Singular Arcs

In the preceding section we addressed the DAE system

$$\dot{\boldsymbol{y}} = \boldsymbol{f}[\boldsymbol{y}, \boldsymbol{u}, t] \tag{48}$$

$$0 = \boldsymbol{g}[\boldsymbol{y}, \boldsymbol{u}, t] \tag{49}$$

which can appear when path constraints are imposed on the optimal control problem. However, even in the absence of path constraints the necessary conditions (31), (36), and (37) lead to the DAE system

$$\dot{\boldsymbol{y}} = \boldsymbol{f}[\boldsymbol{y}, \boldsymbol{u}, t] \tag{50}$$

$$\dot{\boldsymbol{\lambda}} = -\boldsymbol{H}_y^\top \tag{51}$$

$$0 = \boldsymbol{H}_u^\top \tag{52}$$

Viewed as a system of DAEs, one expects the optimality condition $0 = \boldsymbol{H}_u^\top$ to define the control variable provided the matrix $\boldsymbol{H}_{uu}$ is nonsingular. On the other hand, if $\boldsymbol{H}_{uu}$ is a singular matrix, the control $\boldsymbol{u}$ is not uniquely defined by the optimality condition. This situation is referred to as a singular arc, and the analysis of this problem involves techniques quite similar to those discussed earlier for path constraints. Furthermore, singular arc problems are not just mathematical curiosities, because $\boldsymbol{H}_{uu}$ is singular whenever $\boldsymbol{f}[\boldsymbol{y}, \boldsymbol{u}, t]$ is a linear function of $\boldsymbol{u}$. The famous sounding rocket problem proposed by Goddard in 1919 (Ref. 26) contains a singular arc. Recent interest in periodic optimal flight[27,28] and the analysis of wind shear during landing[29] all involve formulations with singular arcs.

D. Algebraic Inequality Constraints

The preceding sections have addressed the treatment of equality path constraints. Let us now consider inequality path constraints of the form

$$0 \le \boldsymbol{g}[\boldsymbol{y}(t), \boldsymbol{u}(t), t] \tag{53}$$

Unlike an equality constraint that must be satisfied for all $t_0 \le t \le t_f$, inequality constraints may either be active $0 = \boldsymbol{g}$ or inactive $0 < \boldsymbol{g}$ at each instant in time. In essence the time domain is partitioned into constrained and unconstrained subarcs. During the unconstrained arcs the necessary conditions are given by Eqs. (31), (36), and (37), whereas the conditions with modified Hamiltonian (44) are applicable in the constrained arcs. Thus the imposition of inequality constraints presents three major complications. First, the number of constrained subarcs present in the optimal solution is not known a priori. Second, the location of the junction points when the transition from constrained to unconstrained (and vice versa) occurs is unknown. Finally, at the junction points it is possible that both the control variables $\boldsymbol{u}$ and the adjoint variables $\boldsymbol{\lambda}$ are discontinuous. Additional jump conditions that are essentially boundary conditions imposed at the junction points must be satisfied. Thus what was a two-point boundary value problem may become a multipoint boundary value problem when inequalities are imposed. For a more complete discussion of this subject, the reader is referred to the tutorial by Pesch[30] and the textbook by Bryson and Ho.[23]

E. Nonlinear Programming vs Optimal Control

To conclude the discussion let us reemphasize the relationship between optimal control and nonlinear programming problems with a simple example. Suppose we must choose the control functions $\boldsymbol{u}(t)$ to minimize

$$J = \phi[\boldsymbol{y}(t_f), t_f] \tag{54}$$

subject to the state equations

$$\dot{\boldsymbol{y}} = \boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t)] \tag{55}$$

where $\boldsymbol{y}(t_0) = \boldsymbol{y}_0$ are given at the fixed initial and final times t_0 and t_f. Let us define the NLP variables

$$\boldsymbol{x} = (\boldsymbol{u}_0, \boldsymbol{y}_1, \boldsymbol{u}_1, \boldsymbol{y}_2, \boldsymbol{u}_2, \ldots, \boldsymbol{y}_M, \boldsymbol{u}_M) \tag{56}$$

as the values of the state and control evaluated at $t_0, t_1, \ldots, t_M$, where $t_k = t_{k-1} + h$ with $h = t_f/M$. Now

$$\dot{\boldsymbol{y}} \approx \frac{\boldsymbol{y}_k - \boldsymbol{y}_{k-1}}{h} \tag{57}$$

Let us substitute this approximation into Eq. (55), thereby defining the NLP constraints

$$c_k(\boldsymbol{x}) = \boldsymbol{y}_k - \boldsymbol{y}_{k-1} - h\boldsymbol{f}(\boldsymbol{y}_{k-1}, \boldsymbol{u}_{k-1}) \tag{58}$$

for $k = 1, \ldots, M$, and the NLP objective function

$$F(\boldsymbol{x}) = \phi(\boldsymbol{y}_M) \tag{59}$$

The problem defined by Eqs. (56), (58), and (59) is a nonlinear program. From Eq. (19) the Lagrangian is

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= F(\boldsymbol{x}) - \boldsymbol{\lambda}^\top \boldsymbol{c}(\boldsymbol{x}) \\ &= \phi(\boldsymbol{y}_M) - \sum_{k=1}^{M} \boldsymbol{\lambda}_k^\top [\boldsymbol{y}_k - \boldsymbol{y}_{k-1} - h\boldsymbol{f}(\boldsymbol{y}_{k-1}, \boldsymbol{u}_{k-1})] \end{aligned} \tag{60}$$

The necessary conditions for this problem follow directly from the definitions (20) and (21):

$$\frac{\partial L}{\partial \boldsymbol{\lambda}_k} = \boldsymbol{y}_k - \boldsymbol{y}_{k-1} - h\boldsymbol{f}(\boldsymbol{y}_{k-1}, \boldsymbol{u}_{k-1}) = 0 \tag{61}$$

$$\frac{\partial L}{\partial \boldsymbol{y}_k} = (\boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k) + h\boldsymbol{\lambda}_{k+1}^\top \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}_k} = 0 \tag{62}$$

$$\frac{\partial L}{\partial \boldsymbol{u}_k} = h\boldsymbol{\lambda}_{k+1}^\top \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_k} = 0 \tag{63}$$

$$\frac{\partial L}{\partial \boldsymbol{y}_M} = -\boldsymbol{\lambda}_M + \frac{\partial \phi}{\partial \boldsymbol{y}_M} = 0 \tag{64}$$

Now let us consider the limiting form of this problem as $M \to \infty$ and $h \to 0$. Clearly, in the limit Eq. (61) becomes the state equation (31), Eq. (62) becomes the adjoint equation (36), Eq. (63) becomes the control equation (37), and Eq. (64) becomes the transversality condition (38). Essentially, what has been demonstrated is that the NLP necessary conditions, i.e., Kuhn–Tucker, approach the optimal control necessary conditions as the number of variables grows. The NLP Lagrange multipliers can be interpreted as discrete approximations to the optimal control adjoint variables. Although this discussion is of theoretical importance, it also suggests a number of ideas that are the basis of modern numerical methods. In particular, if the analysis is extended to inequality constrained problems, it is apparent that the task of identifying the NLP active set is equivalent to defining constrained subarcs and junction points in the optimal control setting. Early results on this transcription process can be found in Refs. 31–33, and more recently interest has focused on using alternate methods of discretization.[34–37]

V. Numerical Analysis

A. Initial Value Problems

The numerical solution of the initial value problem (IVP) for ordinary differential equations (ODE) is fundamental to most trajectory optimization methods. The problem can be stated as follows: Compute the value of $\boldsymbol{y}(t_f)$ for some value of $t_0 < t_f$ that satisfies

$$\dot{\boldsymbol{y}} = \boldsymbol{f}[\boldsymbol{y}(t), t] \tag{65}$$

with the known initial value $\boldsymbol{y}(t_0) = \boldsymbol{y}_0$. Notice that, unlike the state equations (2), the right-hand sides of these equations do not explicitly involve either the controls $\boldsymbol{u}(t)$ or the parameters $\boldsymbol{p}$. This distinction is extremely important in the context of trajectory optimization because this requires that the control is completely determined by specifying the state; i.e., it implies that we can write $\boldsymbol{u}(t) = \tilde{\boldsymbol{g}}[\boldsymbol{y}(t), \boldsymbol{p}, t]$. Numerical methods for solving the ODE IVP are relatively mature in comparison to the other fields in trajectory optimization.

Most schemes can be classified as one-step or multistep methods. A popular family of one-step methods is the Runge–Kutta schemes,

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h_i \sum_{j=1}^{k} \beta_j \boldsymbol{f}_{ij} \tag{66}$$

where

$$\boldsymbol{f}_{ij} = \boldsymbol{f}\left[\left(\boldsymbol{y}_i + h_i \sum_{l=1}^{k} \alpha_{jl} \boldsymbol{f}_{il}\right), (t_i + h_i \rho_j)\right] \tag{67}$$

for $1 \le j \le k$, and k is referred to as the stage. In these expressions $\{\rho_j, \beta_j, \alpha_{jl}\}$ are known constants with $0 \le \rho_1 \le \rho_2 \le \cdots \le 1$. The schemes are called explicit if $\alpha_{jl} = 0$ for $l \ge j$ and implicit otherwise. A convenient way to define the coefficients is to use the so-called Butcher diagram

$$\begin{array}{c|ccc} \rho_1 & \alpha_{11} & \cdots & \alpha_{1k} \\ \vdots & \vdots & & \vdots \\ \rho_k & \alpha_{k1} & \cdots & \alpha_{kk} \\ \hline & \beta_1 & \cdots & \beta_k \end{array}$$

Four common examples of k-stage Runge–Kutta schemes are the following.

Euler's explicit, $k = 1$:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Classical Runge–Kutta explicit, $k = 4$:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Trapezoidal implicit, $k = 2$:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Hermite–Simpson implicit, $k = 3$:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

An obvious appeal of an explicit scheme is that the computation of each integration step can be performed without iteration; that is, given the value $\boldsymbol{y}_i$ at the time t_i, the value $\boldsymbol{y}_{i+1}$ at the new time t_{i+1} follows directly from available values of the right-hand-side functions $\boldsymbol{f}$. In contrast, for an implicit method, the unknown value $\boldsymbol{y}_{i+1}$ appears nonlinearly, e.g., the trapezoidal method requires

$$0 = \boldsymbol{y}_{i+1} - \boldsymbol{y}_i - (h_i/2)[\boldsymbol{f}(\boldsymbol{y}_{i+1}, t_{i+1}) + \boldsymbol{f}(\boldsymbol{y}_i, t_i)] \doteq \boldsymbol{\zeta}_i \tag{68}$$

Consequently, to compute $\boldsymbol{y}_{i+1}$, given the values t_{i+1}, $\boldsymbol{y}_i$, t_i, and $\boldsymbol{f}[\boldsymbol{y}_i, t_i]$, requires solving the nonlinear expression (68) to drive the defect $\boldsymbol{\zeta}_i$ to zero. The iterations required to solve this equation are called corrector iterations. An initial guess to begin the iteration is usually provided by the so-called predictor step. There is considerable latitude in the choice of predictor and corrector schemes. For some well-behaved differential equations, a single predictor and corrector step are adequate. In contrast, it may be necessary to perform multiple corrector iterations, e.g., using Newton's method, especially when the differential equations are stiff. To illustrate this, suppose that instead of Eq. (65) the system dynamics are described by

$$\dot{\boldsymbol{y}} = \boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t), t], \qquad \epsilon\dot{\boldsymbol{u}} = \boldsymbol{g}[\boldsymbol{y}(t), \boldsymbol{u}(t), t] \tag{69}$$

where ϵ is a small parameter. Within a very small region $0 \le t \le t_\epsilon$, the solution displays a rapidly changing behavior, and thereafter the second equation can effectively be replaced by its limiting form

$$0 = \boldsymbol{g}[\boldsymbol{y}(t), \boldsymbol{u}(t), t] \tag{70}$$

The singular perturbation problem (69) is in fact a stiff system of ODEs and in the limit approaches a DAE system. Techniques designed specifically for solving singular perturbation formulations have been suggested for guidance applications.[38,39]

The second class of integration schemes are termed multistep schemes and have the general form

$$\boldsymbol{y}_{i+k} = \sum_{j=0}^{k-1} \alpha_j \boldsymbol{y}_{i+j} + h \sum_{j=0}^{k} \beta_j \boldsymbol{f}_{i+j} \tag{71}$$

where α_j and β_j are known constants. If $\beta_k = 0$, the method is explicit; otherwise it is implicit. The Adams schemes are members of the multistep class that are based on approximating the functions $\boldsymbol{f}(t)$ by interpolating polynomials. The Adams–Bashforth method is an explicit multistep method,[40] whereas the Adams–Moulton method is implicit.[41] Multistep methods must address three issues that we have not discussed for single-step methods. First, as written, the method requires information at $(k - 1)$ previous points. Clearly, this implies some method must be used to start the process, and one common technique is to take one or more steps with a one-step method, e.g., Euler. Second, as written, the multistep formula assumes the stepsize h is a fixed value. When the stepsize is allowed to vary, careful implementation is necessary to ensure that the calculation of the coefficients is both efficient and well conditioned. Finally, similar remarks apply when the number of steps k, i.e., the order, of the method is changed.

Regardless of whether a one-step or a multistep method is utilized, a successful implementation must address the accuracy of the solution. How well does the discrete solution $\boldsymbol{y}_i$ for $i = 0, 1, \ldots, M$ produced by the integration scheme agree with the real answer $\boldsymbol{y}(t)$? All well-implemented schemes have some mechanism for adjusting the integration step size and/or order to control the integration error. The reader is urged to consult Refs. 42–46 for additional information. A great deal of discussion has been given to the distinction between explicit and implicit methods. Indeed, it is often tempting to use an explicit method simply because it is more easily implemented (and understood). However, the trajectory optimization problem is a boundary value problem (BVP), not an initial value problem, and to quote Ascher et al. on page 69 of Ref. 24, "for a boundary value problem . . . any scheme becomes effectively implicit. Thus, the distinction between explicit and implicit initial value schemes becomes less important in the BVP context."

Methods for solving initial value problems when dealing with a system of differential-algebraic equations have appeared more recently. For a semiexplicit DAE system such as Eqs. (48) and (49), it is tempting to try to eliminate the algebraic (control) variables to utilize a more standard method for solving ODEs. Proceeding formally to solve Eq. (49), one can write

$$\boldsymbol{u}(t) = \boldsymbol{g}^{-1}[\boldsymbol{y}, t] \tag{72}$$

When this value is substituted into Eq. (48), one obtains the nonlinear differential equation

$$\dot{\boldsymbol{y}} = \boldsymbol{f}\{\boldsymbol{y}, \boldsymbol{g}^{-1}[\boldsymbol{y}, t], t\} \tag{73}$$

which is amenable to solution using any of the ODE techniques described earlier. Another elimination technique, referred to as differential inclusion,[47] attempts to form an expression of the form

$$\boldsymbol{u}(t) = \mathcal{F}[\dot{\boldsymbol{y}}, \boldsymbol{y}, t] \tag{74}$$

by solving a subset of the differential equations (48). Because the number of state and control variables is not necessarily equal, it is imperative to partition the differential equations in some stable manner to perform this elimination. Unfortunately, it is seldom possible to analytically construct a feedback control of the form of Eq. (72) or Eq. (74). When analytic elimination is impossible, the only recourse is to introduce a nonlinear iterative technique, e.g., Newton's method, that must be executed at every integration step. This approach not only is very time consuming but also can conflict with logic used to control integration error in the dynamic variables $\boldsymbol{y}$. If an implicit method is used for solving the ODEs, this elimination iteration must be performed within each corrector iteration; in other words, it becomes an iteration within an iteration. In essence, methods that attempt to eliminate the control to avoid the DAE problem are cumbersome, numerically unstable, and problem specific.

The first general technique for solving DAEs was proposed by Gear[48] and utilizes a backward differentiation formula (BDF) in a linear multistep method. In contrast to the elimination methods in the preceding paragraph, the algebraic variables $\boldsymbol{u}(t)$ are treated the same as the differential variables $\boldsymbol{y}(t)$. The method was originally proposed for the semiexplicit index 1 system described by Eqs. (48) and (49) and soon extended to the fully implicit form

$$F[\dot{z}, z, t] = 0 \tag{75}$$

where $z = (y, u)$. The basic idea of the BDF approach is to replace the derivative $\dot{z}$ by the derivative of the polynomial that interpolates the solution computed over the preceding k steps. The simplest example is the implicit Euler method that replaces Eq. (75) with

$$F\left[\frac{z_i - z_{i-1}}{h_i}, z_i, t_i\right] = 0 \tag{76}$$

The resulting nonlinear system in the unknowns z_i is usually solved by some form of Newton's method at each time step t_i. The widely used production code DASSL developed by Petzold[49,50] essentially uses a variable-step-size, variable-order implementation of the BDF formulas. The method is appropriate for index 1 DAEs with consistent initial conditions. Current research into the solution of DAEs with higher index (≥ 2) has renewed interest in one-step methods, specifically the implicit Runge–Kutta schemes described for ODEs. A discussion of methods for solving DAEs is found in Ref. 25.

B. Tabular Data

In practice the numerical solution of a trajectory optimization problem inevitably involves tabular data. Typically, propulsion, aerodynamic, weight, and mass properties for a vehicle are specified using tables. For example, the thrust of a motor may be specified by a finite set of table values $\{T(M_k, h_k), M_k, h_k\}$ for $k = 1, \ldots, N$, in lieu of defining the functional form in terms of Mach number and altitude. In some cases this approach is necessary simply because there is not enough information to permit an analytic representation of the function based on the laws of physics. Often tabular data are obtained as the result of experimental tests. Finally, there may be historical precedence for specifying data in this format as a convenient way for communication between disciplines. Regardless of the reason for specifying a nonlinear function as a collection of tabular values, the numerical implementation of a trajectory optimization must deal with this format. The necessary conditions described in Sec. III assume continuity and differentiability for the objective and constraint functions. Similar restrictions are implied when stating the necessary conditions for the optimal control problem in Sec. IV. The numerical integration techniques given in the previous subsection make similar assumptions about continuity and differentiability for the right-hand sides of the differential algebraic equations. Successful application of those techniques requires that tabular data be represented using a smooth differentiable function. Unfortunately, by far the single most widely used approach is linear interpolation. This is also by far the single most catastrophic impediment to an efficient solution of the trajectory optimization problem! A piecewise linear representation is not differentiable at the table points and thus is fundamentally inconsistent with the theory described in the preceding sections. Recognition of this difficulty is certainly not new, and it has been discussed by other authors[51,52] in the simulation of space launch vehicles. Nevertheless, inappropriate data modeling techniques persist, presumably for historical reasons, in many real applications.

There are many alternatives for representing tabular data using a smooth functional form. For some applications an appropriate model is suggested by the physics, e.g., a quadratic drag polar. In lieu of a form derived from physical considerations, functional approximation based solely on the mathematical requirements can be incorporated. Function approximation using B-splines[53] can effectively produce the required continuity. Methods for constructing smooth approximations utilizing nonlinear programming techniques have also been developed.[54–57] Nonlinear rational function or neural network approximations can also produce sufficient smoothness, although it is not clear that these are preferable to B-splines.

VI. Compendium of Methods

The basic elements involved in the specification of a numerical method for solving the trajectory optimization problem have been described in the preceding sections. There is a broad spectrum of possible ways to put the pieces together to form a complete algorithm; however, all techniques have one attribute in common. Because all of the algorithms involve application of Newton's method, a convenient way to organize the discussion is to describe the function evaluation procedure for each method. Specifically, we will describe the function generator for each algorithm. The inputs to the function generator are the variables. The outputs of the function generator are the objective and constraints. The basic concept is illustrated in Fig. 1.

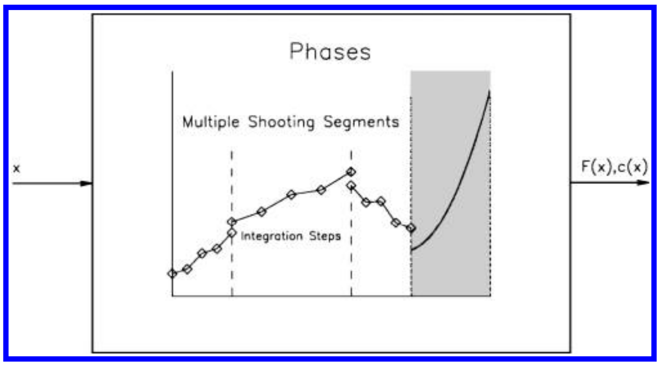

Fig. 1 Function generator.

A. Direct Shooting

1. Algorithm

The variables for a direct shooting application are chosen as a subset of the initial conditions, the final conditions, and the parameters. Thus for each phase let us define

$$X^{(k)} = \{y(t_0), p, t_0, y(t_f), t_f\} \tag{77}$$

The total set of NLP variables is then

$$x \subset \left\{X^{(1)}, X^{(2)}, \ldots, X^{(N)}\right\} \tag{78}$$

Notice that any time-varying quantities must be represented using the finite set of parameters x, and consequently this implies that the control/time history must be defined by a finite set of parameters. For example, one might have an explicit representation such as

$$u = p_1 + p_2 t \tag{79}$$

or an implicit relationship such as

$$0 = p_1 u(t) + \sin[p_2 u(t)] \tag{80}$$

When the control is defined explicitly as in Eq. (79), propagation of the trajectory from the beginning to the end of the phase can be accomplished using an ODE initial value method as described. On the other hand, if the control is defined implicitly, the phase propagation will require the use of a DAE initial value method as described in Sec. V.A. Notice also that problems with path inequality constraints (5) must be treated as a sequence of constrained and unconstrained arcs. Thus phases must be introduced to account for these individual arcs, in addition to phases that are necessary to model known problem discontinuities such as jettison of a stage.

The NLP constraints and objective function are quantities that are evaluated at the boundaries of one or more of the phases. Thus we have

$$c(x) = \begin{bmatrix} \psi^{(1)}[y(t_0), p, t_0] \\ \psi^{(1)}[y(t_f), p, t_f] \\ \vdots \\ \psi^{(N)}[y(t_0), p, t_0] \\ \psi^{(N)}[y(t_f), p, t_f] \end{bmatrix} \tag{81}$$

In summary, the function generator for the direct shooting method is of the following form:

Direct Shooting

Input: $\boldsymbol{x}$
Do for (each phase) $k = 1, N$
  Initialize phase k: $\boldsymbol{y}^{(k)}(t_0)$, $\boldsymbol{p}^{(k)}$, $t_0^{(k)}$
  Constraint evaluation: compute $\psi^{(k)}[\boldsymbol{y}(t_0), \boldsymbol{p}, t_0]$
  Initial value problem: given t_f, compute
    $\boldsymbol{y}^{(k)}(t_f)$, i.e., solve Eq. (65) or Eqs. (48) and (49)
  Constraint evaluation: compute $\psi^{(k)}[\boldsymbol{y}(t_f), \boldsymbol{p}, t_f]$
End do
Terminate trajectory
  Compute objective $F(\boldsymbol{x}), \boldsymbol{c}(\boldsymbol{x})$
Output: $F(\boldsymbol{x}), \boldsymbol{c}(\boldsymbol{x})$

2. Examples

The direct shooting method is one of the most widely used methods and is especially effective for launch vehicle and orbit transfer applications. The program to optimize simulated trajectories (POST) program[58] developed by Martin Marietta for simulating the trajectories of launch vehicles such as the Titan is a widely distributed implementation of the direct shooting method. Originally developed to support military space applications, it is similar in functionality to the generalized trajectory simulation (GTS) program[59] developed at The Aerospace Corporation. Most major aerospace firms either use POST or have an equivalent capability for launch vehicle optimization and mission analysis. Early versions of POST utilized a reduced gradient optimization algorithm similar to the methods in Refs. 9 and 10, and more recent releases have incorporated an SQP method.[7] GTS utilizes a modified form of the reduced gradient algorithm,[8] which incorporates quasi-Newton updates for constraint elimination and Hessian approximation. Programs such as POST and GTS have extensive libraries of application-specific models. In particular, the libraries permit definition of the vehicle dynamics {the right-hand-side functions $\boldsymbol{f}[\boldsymbol{y}(t), \boldsymbol{u}(t), \boldsymbol{p}, t]$} in many coordinate systems, e.g., Earth-centered inertial, intrinsic, orbital, etc. It is also common to have 10–20 different models for computing the gravitational, propulsive, and aerodynamic forces. In most cases the user can also specify the type of numerical integration and interpolation to be used, as well as the trajectory input and output formats.

Direct shooting applications have been most successful in launch and orbit transfer problems primarily because this class of problem lends itself to parameterization with a relatively small number of NLP variables. For example, an orbit transfer problem with impulsive burns[60,61] can be posed with four variables per burn, namely, the time of ignition and the velocity increment $(t_i, \Delta \boldsymbol{V}_i)$. Typically the mission orbit can be defined using three to five nonlinear constraints enforced at the end of the trajectory. Thus a typical two-burn orbit transfer can be posed as an NLP with eight variables and four or five constraints. When the vehicle thrust-to-weight ratio is high, there is little motivation to consider a more elaborate mathematical model of the thrust variation for two reasons. First, the performance benefit that can be achieved with a thrust variation, i.e., by introducing time-varying control $\boldsymbol{u}(t)$, is negligible. Second, most real vehicles do not have the ability to implement a variable direction thrust even if it was computed. In fact, many spacecraft incorporate spin stabilization, which implies a constant inertial attitude during the burns. Stated simply, the application neither permits nor warrants a mathematical model of higher fidelity, and direct shooting is very effective.

A similar situation exists when designing optimal launch vehicle trajectories. During the early portion of an ascent trajectory, it is common to define the turning by a finite set of pitch rates. This approach is used for most expendable launch vehicles, e.g., Titan, Delta, and Atlas/Centaur, and is often a part of the onboard mission data load. Consequently, the steering during the early portion of a launch vehicle trajectory is defined by a relatively small number of parameters, and the resulting optimization problem is readily formulated using the direct shooting method. Steering during the second stage of the Space Shuttle ascent trajectory is defined by a linear tangent steering law. In this case the control can be defined by six parameters $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$:

$$\boldsymbol{u}(t) = \boldsymbol{p}_1 + \boldsymbol{p}_2 t \tag{82}$$

which then determine the inertial yaw and pitch angles according to

$$\psi_I = \arctan\{u_2/u_1\} \tag{83}$$

$$\theta_I = \arcsin\{u_3/\|\boldsymbol{u}\|\} \tag{84}$$

This form of the control law is an exact solution of the optimal control problem when gravity is constant[23,62] and is implemented in the Space Shuttle's flight avionics. Again the optimal steering is approximated by a finite set of parameters, and the resulting trajectory optimization problem is amenable to direct shooting.

3. Issues

Most successful direct shooting applications have one salient feature in common, namely, the ability to describe the problem in terms of a relatively small number of optimization variables. If the dynamic behavior of the control functions $\boldsymbol{u}(t)$ cannot be represented using a limited number of NLP variables, the success of a direct shooting method can be degraded significantly. For example, it is tempting to approximate the controls using an expansion such as

$$\boldsymbol{u}(t) = \sum_{k=1}^{M} \boldsymbol{p}_k B_k(t) \tag{85}$$

where $M \gg 1$, and $B_k(t)$ are as a set of basis functions, e.g., B-spline. This approach impacts the direct shooting method in two ways. Both are related to the calculation of gradient information for the NLP iteration. The first issue is related to the sensitivity of the variables. Changes early in the trajectory (near t_0) propagate to the end of the trajectory. The net effect is that the constraints can behave very nonlinearly with respect to variables, thereby making the optimization problem difficult to solve. This is one of the major reasons for the multiple shooting techniques, which will be described later. The second issue is the computational cost of evaluating the gradient information. The most common approach to computing gradients is via finite difference approximations. A forward difference approximation to column j of the Jacobian matrix $\boldsymbol{G}$ in Eq. (20) is

$$\boldsymbol{G}_{\cdot j} = (1/\delta_j)[\boldsymbol{c}(\boldsymbol{x} + \boldsymbol{\delta}_j) - \boldsymbol{c}(\boldsymbol{x})] \tag{86}$$

where the vector $\boldsymbol{\delta}_j = \delta_j \boldsymbol{e}_j$, and $\boldsymbol{e}_j$ is a unit vector in direction j. A central difference approximation is

$$\boldsymbol{G}_{\cdot j} = (1/2\delta_j)[\boldsymbol{c}(\boldsymbol{x} + \boldsymbol{\delta}_j) - \boldsymbol{c}(\boldsymbol{x} - \boldsymbol{\delta}_j)] \tag{87}$$

To calculate gradient information this way, it is necessary to integrate the trajectory for each perturbation. Consequently, at least n trajectories are required to evaluate a finite difference gradient, and this information may be required for each NLP iteration. The cost of computing the finite difference gradients is reduced somewhat in the GTS[59] program by using a partial trajectory mechanism. This approach recognizes that it is not necessary to integrate the trajectory from t_0 to t_f if the optimization variable is introduced later, say at $t_s > t_0$. Instead the gradient information can be computed by integrating from the return point t_r to t_f, where $t_s \geq t_r > t_0$, because the portion of the trajectory from t_0 to t_r will not be altered by the perturbation. A less common alternative to finite difference gradients is to integrate the so-called variational equations. In this technique, an additional differential equation is introduced for each NLP variable, and this augmented system of differential equations must be solved along with the state equations. Unfortunately, the variational equations must be derived for each application and consequently are used far less in general purpose trajectory software.

Another issue that must be addressed is the accuracy of the gradient information. Forward difference estimates are of order δ, whereas central difference estimates are $\mathcal{O}(\delta^2)$. Of course the more accurate central difference estimates are twice as expensive as forward difference gradients. Typically, numerical implementations use forward difference estimates until nearly converged and then switch to the more accurate derivatives for convergence. Although techniques for selecting the finite difference perturbation size might seem to be critical to accurate gradient evaluation, a number of effective methods are available to deal with this matter.[4] A more crucial matter is the interaction between the gradient computations and the underlying numerical interpolation and integration algorithms. We have already discussed how linear interpolation of tabular data can introduce gradient errors. However, it should be emphasized that sophisticated predictor corrector variable-step/variable-order numerical integration algorithms also introduce noise into the gradients. Although those techniques enhance the efficiency of the integration, they degrade the efficiency of the optimization. In fact, a simple fixed-step/fixed-order integrator may yield better overall efficiency in the trajectory optimization because the gradient information is more accurate. Two integration methods that are suitable for use inside the trajectory function generator are described in Ref. 63 and Refs. 64 and 65. Another issue arises in the context of a trajectory optimization application when the final time t_f is defined implicitly, not explicitly, by a boundary or event condition. In this case we are not asking to integrate from t_0 to t_f but rather from t_0 until $\psi[\boldsymbol{y}(t_f), t_f] = 0$. Most numerical integration schemes interpolate the solution to locate the final point. On the other hand, if the final point is found by iteration, e.g., using a root-finding method, the net effect is to introduce noise into the external Jacobian evaluations. A better alternative is to simply add an extra variable and constraint to the overall NLP problem and avoid the use of an internal iteration. In fact, inaccuracies in the gradient can be introduced by 1) internal iterations, e.g., solving Kepler's equation, event detection; 2) interpolation of tabular data; and 3) discontinuous functions, e.g., "ABS," "MAX," "IF" tests; and a carefully implemented algorithm must avoid these difficulties.[66]

B. Indirect Shooting

1. *Algorithm*

Let us begin with a description of indirect shooting for the simplest type of optimal control problem with no path constraints and a single phase. The variables are chosen as a subset of the boundary values for the optimal control necessary conditions. For this case the NLP variables are

$$\boldsymbol{x} = \{\boldsymbol{\lambda}(t_0), t_f\} \tag{88}$$

and the NLP constraints are

$$\boldsymbol{c}(\boldsymbol{x}) = \begin{bmatrix} \psi[\boldsymbol{y}(t), \boldsymbol{p}, t] \\ \boldsymbol{\lambda}(t) - \boldsymbol{\Phi}_y^\top \\ (\Phi_t + H) \end{bmatrix}\Bigg|_{t=t_f} \tag{89}$$

A major difference between direct and indirect shooting occurs in the definition of the control functions $\boldsymbol{u}(t)$. For indirect shooting the control is defined at each point in time by the maximum principle (41) or (37). Thus in some sense the values $\boldsymbol{\lambda}(t_0)$ become the parameters that define the optimal control function instead of $\boldsymbol{p}$. When the maximum principle is simple enough to permit an explicit definition of the control, propagation of the trajectory from the beginning to the end of the phase can be accomplished using an ODE initial value method. On the other hand, if the control is defined implicitly, the phase propagation will require the use of a DAE initial value method as described in Sec. V.A. Notice also that problems with path inequality constraints (5) must be treated as a sequence of constrained and unconstrained arcs. Thus phases must be introduced to account for these individual arcs just as with direct shooting. When additional phases are introduced, in general it will be necessary to augment the set of variables to include the unknown adjoint and multipliers at each of the phase boundaries. Furthermore, additional constraints are included to reflect the additional necessary conditions.

In summary, the function generator for the indirect shooting method is of the following form:

Indirect Shooting

Input: $\boldsymbol{x}$
Do for (each phase) $k = 1, N$
  Initialize phase k: $\boldsymbol{y}^{(k)}(t_0)$, $\boldsymbol{\lambda}^{(k)}(t_0)$, $\boldsymbol{p}^{(k)}$, $t_0^{(k)}$
  Initial value problem: given t_f compute $\boldsymbol{y}^{(k)}(t_f)$, $\boldsymbol{\lambda}^{(k)}(t_f)$, i.e., solve Eqs. (31), (36), and (37)
  Constraint evaluation: evaluate Eq. (89)
End do
Terminate trajectory
Output: $\boldsymbol{c}(\boldsymbol{x})$

2. *Examples*

Although the indirect shooting method would seem to be quite straightforward, it suffers from a number of difficulties that will be described in the next section. Primarily because of the computational limitations, successful applications of indirect shooting have focused on special cases. Because the method is very sensitive to the initial guess, it is most successful when the underlying dynamics are rather benign. The method has been utilized for launch vehicle trajectory design in the program DUKSUP[67] and for low-thrust orbit analysis.[68]

3. *Issues*

The sensitivity of the indirect shooting method has been recognized for some time. Computational experience with the technique in the late 1960s is summarized by Bryson and Ho on page 214 of Ref. 23:

> The main difficulty with these methods is *getting started*; i.e., finding a first estimate of the unspecified conditions at one end that produces a solution reasonably close to the specified conditions at the other end. The reason for this peculiar difficulty is the extremal solutions are often *very sensitive* to small changes in the unspecified boundary conditions. . . . Since the system equations and the Euler–Lagrange equations are coupled together, it is not unusual for the numerical integration, with poorly guessed initial conditions, to produce "wild" trajectories in the state space. These trajectories may be so wild that values of $x(t)$ and/or $\lambda(t)$ exceed the numerical range of the computer!

A number of techniques have been proposed for dealing with this sensitivity. One rather obvious approach is to begin the iteration process with a good initial guess. Referred to as imbedding, continuation, or homotopy methods, the basic idea is to solve a sequence of problems and use the solution of one problem as the initial guess for a slightly modified problem. Thus, suppose it is necessary to solve $\boldsymbol{a}(\boldsymbol{x}) = 0$ and that we can imbed this problem into a family of related problems

$$\bar{\boldsymbol{a}}(\boldsymbol{x}, \tau) = 0 \tag{90}$$

where the parameter $0 \le \tau \le 1$. Assume the problem $\bar{\boldsymbol{a}}(\boldsymbol{x}, 0) = 0$ is easy to solve, and when $\tau = 1$, the real solution is obtained, i.e., $\bar{\boldsymbol{a}}(\boldsymbol{x}, 1) = \boldsymbol{a}(\boldsymbol{x}) = 0$. Typically, it is desirable to choose the parameter τ such that the solution $\boldsymbol{x}(\tau)$ varies smoothly along the homotopy path. For example, it may be easy to solve an orbit transfer using two-body dynamics. A more accurate solution involving oblate Earth perturbations could then be obtained by turning on the gravitational perturbations with an imbedding technique. Clearly, the homotopy method can be used with any trajectory optimizing algorithm, but it has been especially useful for indirect methods.

Another technique that has been used to reduce the solution sensitivity is referred to as the sweep method. Essentially, the idea is to integrate the state equations forward, i.e., from t_0 to t_f, and then integrate the adjoint equations backward, i.e., from t_f to t_0. The goal is to exploit the fact that the state equations may be integrated stably in the forward direction and the adjoint equations may be stable in the reverse direction. This approach requires that the state, control, and adjoint time histories be saved using some type of interpolation method. Numerical processing is further complicated by interaction between the interpolation scheme and the integration error control, especially when dealing with discontinuities that occur at phase boundaries.

Perhaps the biggest issue that must be addressed when using an indirect method is the derivation of the necessary conditions themselves. For realistic trajectory simulations the differential equations (2), path constraints (5), and boundary conditions (4) may all be complicated mathematical expressions. To impose the optimality conditions (36–39) it is necessary to analytically differentiate the expressions for $\boldsymbol{f}$, $\boldsymbol{g}$, and $\boldsymbol{\psi}$. For a production software tool such as POST or GTS, which permits problem formulation using alternate coordinate systems and propulsion, gravitational, and aerodynamic models, this can be a daunting task! As a consequence, the optimality conditions are usually not derived for all possible combinations and models. The impact is that current implementations of indirect methods suffer from a lack of flexibility. Let us emphasize that this is a limitation in current but not necessarily future implementations. In particular, a number of authors have explored the utility of automatic differentiation tools to eliminate this impediment. For example, the ADIFOR software,[69] the OCCAL software,[70] and the approach taken by Mehlhorn and Sachs[71] represent promising attempts to automate this process.

C. Multiple Shooting

1. *Algorithm*

Both the direct and indirect shooting methods suffer from a common difficulty. The essential shortcoming of these methods is that small changes introduced early in the trajectory can propagate into very nonlinear changes at the end of the trajectory. Although this effect can be catastrophic for an indirect method, it also represents a substantial limitation on the utility of a direct formulation. The basic notion of multiple shooting (in contrast to simple shooting) was originally introduced[72,73] for solving two-point boundary value problems, and we begin the discussion for this case. In its simplest form the problem can be stated as follows: Compute the unknown initial values $\boldsymbol{v}(t_0) = \boldsymbol{v}_0$ such that the boundary condition

$$0 = \phi[\boldsymbol{v}(t_f), t_f] \tag{91}$$

holds for some value of $t_0 < t_f$ that satisfies

$$\dot{\boldsymbol{v}} = \boldsymbol{f}[\boldsymbol{v}(t), t] \tag{92}$$

The fundamental idea of multiple shooting is to break the trajectory into shorter pieces or segments. Thus we break the time domain into smaller intervals of the form

$$t_0 < t_1 < \cdots < t_M = t_f \tag{93}$$

Let us denote $\boldsymbol{v}_j$ for $j = 0, \ldots, (M-1)$, as the initial value for the dynamic variable at the beginning of each segment. For segment j we can integrate the differential equations (92) from t_j to the end of the segment at t_{j+1}. Denote the result of this integration by $\bar{\boldsymbol{v}}_j$. Collecting the results for all segments, let us define a set of NLP variables

$$\boldsymbol{x} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{M-1}\} \tag{94}$$

Now we also must ensure that the segments join at the boundaries; consequently, we impose the constraints

$$\boldsymbol{c}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{v}_1 - \bar{\boldsymbol{v}}_0 \\ \boldsymbol{v}_2 - \bar{\boldsymbol{v}}_1 \\ \vdots \\ \phi[\boldsymbol{v}_M, t_f] \end{bmatrix} = 0 \tag{95}$$

One obvious result of the multiple shooting approach is an increase in the size of the problem that the Newton iteration must solve since additional variables and constraints are introduced for each shooting segment. In particular the number of NLP variables and constraints for a multiple shooting application is $n = n_v M$ where n_v is the number of dynamic variables $\boldsymbol{v}$, and M is the number of segments. Fortunately the Jacobian matrix $\boldsymbol{A}$ that appears in the calculation of the Newton search direction (12) is sparse. In particular, only Mn_v^2 elements in $\boldsymbol{A}$ are nonzero. This sparsity is a direct consequence of the multiple shooting formulation because variables early in the trajectory do not change constraints later in the trajectory. In fact, Jacobian sparsity is the mathematical consequence of uncoupling between the multiple shooting segments. For the simple case described, the Jacobian matrix is banded with $n_v \times n_v$ blocks along the diagonal, and very efficient methods for solving the linear system (12) can be utilized. Note that the multiple shooting segments are introduced strictly for numerical reasons. The original optimal control problem may also have phases as described previously. Thus, in general, each phase will be subdivided into multiple shooting segments as illustrated in Fig. 1. Furthermore, within each phase the set of differential-algebraic equations and corresponding boundary conditions may be different, depending on whether the arc is constrained or unconstrained, etc.

The multiple shooting concept can be incorporated into either a direct or indirect method. The distinction between the two occurs in the definition of the dynamic variables $\boldsymbol{v}$, the dynamic system (92), and the boundary conditions (91). For a direct multiple shooting method, we can identify the dynamic variables $\boldsymbol{v}$ with the state and control $(\boldsymbol{y}, \boldsymbol{u})$. By analogy the dynamics are given by the original state equations (2) and path constraints (5). In lieu of the simple boundary conditions (91), we directly impose Eqs. (3) and (4). For an indirect multiple shooting algorithm the dynamic variables $\boldsymbol{v}$ must include the state, control, and adjoint variables $(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\lambda})$. The dynamics are given by the original state equations (2) and the appropriate necessary conditions (36) and (37). In this instance the boundary conditions (91) are replaced with the transversality conditions (38) and (39) along with (3) and (4). It also may be necessary to augment the set of NLP iteration variables $\boldsymbol{x}$ and constraints $\boldsymbol{c}(\boldsymbol{x})$ to account for the additional conditions that occur when entering and leaving path inequalities. As a final distinction for an indirect method, the number of NLP variables $\boldsymbol{x}$ and constraints $\boldsymbol{c}(\boldsymbol{x})$ are equal, i.e., $m = n$, because the optimality conditions uniquely define the values of the NLP variables. For a direct method, n and m may differ, and the objective function $F(\boldsymbol{x})$ must be used to define the optimal values of the NLP variables.

The function generator for the multiple shooting method is of the following form:

Multiple Shooting

Input: $\boldsymbol{x}$
Do for (each phase) $k = 1, N$
  Initialize phase k:
  Do for (each segment) $j = 0, M - 1$
    Initialize segment $j + 1$: $\boldsymbol{v}_j$, t_j
    Initial value problem: given t_{j+1}, compute $\bar{\boldsymbol{v}}_j$,
      i.e., solve DAE system
    Constraint evaluation: save $\boldsymbol{v}_{j+1} - \bar{\boldsymbol{v}}_j$ in Eq. (95)
  End do
    Save $\phi[\boldsymbol{v}_M, t_f]$ in Eq. (95)
End do
Terminate trajectory
Output: $\boldsymbol{c}(\boldsymbol{x})$ [and $F(\boldsymbol{x})$]

2. *Examples*

Perhaps the single most important benefit derived from a multiple shooting formulation (either direct or indirect) is enhanced robustness. The BNDSCO implementation[74] of indirect multiple shooting is widely used in Germany to solve very difficult applications. An optimal interplanetary orbit transfer involving planetary perturbations has been computed by Callies.[75] Berkmann and Pesch[29] have utilized the approach for the study of landing in the presence of windshear, Kreim et al.[76] for Shuttle re-entry, and Pesch[30] for a number of other aerospace applications. System identification problems have been addressed by Bock and Plitt.[77]

An interesting benefit of the multiple shooting algorithm is the ability to exploit a parallel processor. The method is sometimes called parallel shooting, because the simulation of each segment and/or phase can be implemented on an individual processor. This technique was explored for a direct multiple shooting method[78] and remains an intriguing prospect for multiple shooting methods in general.

3. *Issues*

The multiple shooting technique greatly enhances the robustness of either direct or indirect methods. However, the number of NLP iteration variables and constraints increases markedly over simple shooting implementations. Consequently, it is imperative to exploit matrix sparsity to efficiently solve the NLP Newton equations. For indirect shooting, the matrix $\boldsymbol{A}$ has a simple block banded structure, and efficient linear algebra methods are rather straightforward. For direct shooting, sparsity appears both in the Jacobian $\boldsymbol{G}$ and the Hessian $\boldsymbol{H}$, and the relevant sparse linear system is the KT system (22). This system can be solved efficiently using the multifrontal method for symmetric indefinite matrices.[13] In general, all of the other issues associated with simple direct and indirect shooting still apply. Perhaps the most perplexing difficulty with shooting methods is the need to define constrained and unconstrained subarcs a priori when solving problems with path inequalities.

D. Indirect Transcription

1. Algorithm

Historically, transcription or collocation methods were developed for solving two-point boundary value problems such as those encountered when solving the necessary conditions with an indirect formulation. Let us again consider the simple boundary value problem (91) and (92) and, as with multiple shooting, subdivide the interval as in Eq. (93). A fundamental part of the multiple shooting method was to solve the ODEs using an initial value method. Let us consider taking a single step with an explicit method such as Euler's. Following the multiple shooting methodology, we then must impose constraints of the form

$$0 = \boldsymbol{v}_{j+1} - \bar{\boldsymbol{v}}_j \tag{96}$$

$$= \boldsymbol{v}_{j+1} - (\boldsymbol{v}_j + h_j \boldsymbol{f}_j) \tag{97}$$

where $t_{j+1} = h_j + t_j$ for all of the segments $j = 0, \ldots, (M-1)$. Of course, there is no reason to restrict the approximation to an Euler method. In fact, if we incorporate an implicit scheme such as the trapezoidal method (68), satisfaction of the defect constraint (97) is exactly the same as the corrector iteration step described for all implicit integrators. The only difference is that all of the corrector iterations are done at once in the boundary value context, whereas the corrector iterations are done one at a time (step by step) when the process is part of an initial value integration method. One of the most popular and effective choices for the defect constraint in collocation methods is the Hermite–Simpson method[79]:

$$0 = \boldsymbol{v}_{j+1} - \boldsymbol{v}_j - (h_{j+1}/6)[\boldsymbol{f}_{j+1} + 4\bar{\boldsymbol{f}}_{j+1} + \boldsymbol{f}_j] \doteq \boldsymbol{\zeta}_j \tag{98}$$

which is the Simpson defect with Hermite interpolant

$$\bar{\boldsymbol{v}}_{j+1} = \tfrac{1}{2}[\boldsymbol{v}_j + \boldsymbol{v}_{j+1}] + (h_{j+1}/8)[\boldsymbol{f}_j - \boldsymbol{f}_{j+1}] \tag{99}$$

for the variables at the interval midpoint. Schemes of this type are referred to as collocation methods[80] because the solution is a piecewise continuous polynomial that collocates, i.e., satisfies, the ODEs at the so-called collocation points in the subinterval $t_j \le t \le t_{j+1}$. For obvious reasons, the points t_j are also called grid points, mesh points, or nodes.

The function generator for the indirect transcription method is of the following form:

Indirect Transcription

Input: $\boldsymbol{x}$
Do for (each phase) $k = 1, N$
  Initialize phase k:
  Do for (each grid point) $j = 0, M-1$
    Constraint evaluation: save discretization defect, e.g., Eq. (98) or (68)
  End do
    Save $\phi[\boldsymbol{v}_M, t_f]$ in Eq. (95)
End do
Terminate trajectory
Output: $\boldsymbol{c}(\boldsymbol{x})$

2. Examples

Collocation methods have been used for solving boundary value problems occurring in many fields for nearly 40 years, and the reader is urged to consult Refs. 24, 80, and 81. Dickmanns[81] implemented the Hermite–Simpson method in the CHAP3 software and reported successful solution of a number of challenging applications, including Shuttle re-entry problems with convective heating and maximum cross-range capability. The COLSYS package developed by Ascher et al.[82] has also been widely used for boundary value problems.

3. Issues

Collocation methods can be extremely effective for solving multipoint boundary value problems such as those encountered when optimizing a trajectory. As with all indirect methods, however, the techniques cannot be applied without computing the adjoint equations. Furthermore, when path inequality constraints are present, it is imperative to predetermine the sequence of constrained and unconstrained subarcs to formulate the correct BVP.

E. Direct Transcription

1. Algorithm

There are two major reasons that direct transcription methods[83] are actively being investigated. First, like all direct methods, they can be applied without explicitly deriving the necessary conditions, i.e., adjoint, transversality, maximum principle. This feature makes the method appealing for complicated applications and promises versatility and robustness. Second, in contrast to all other techniques described, direct transcription methods do not require an a priori specification of the arc sequence for problems with path inequalities.

Just as before, we break the time domain into smaller intervals of the form

$$t_0 < t_1 < \cdots < t_M = t_f \tag{100}$$

The NLP variables then become the values of the state and control at the grid points, namely,

$$\boldsymbol{x} = \{\boldsymbol{y}_0, \boldsymbol{u}_0, \boldsymbol{y}_1, \boldsymbol{u}_1, \ldots, \boldsymbol{y}_M, \boldsymbol{u}_M\} \tag{101}$$

The set of NLP variables may be augmented to include the parameters $\boldsymbol{p}$, the times t_0 and t_f, and for some discretizations the values of the state and control at collocation points between the grid points. The key notion of the collocation methods is to replace the original set of ODEs (2) with a set of defect constraints $\boldsymbol{\zeta}_i = 0$, which are imposed on each interval in the discretization. Thus, when combined with the original path constraints, the complete set of NLP constraints is

$$\boldsymbol{c}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\psi}_0 \\ \boldsymbol{g}[\boldsymbol{y}_0, \boldsymbol{u}_0, \boldsymbol{p}, t_0] \\ \boldsymbol{\zeta}_0 \\ \boldsymbol{g}[\boldsymbol{y}_1, \boldsymbol{u}_1, \boldsymbol{p}, t_1] \\ \boldsymbol{\zeta}_1 \\ \vdots \\ \boldsymbol{g}[\boldsymbol{y}_{M-1}, \boldsymbol{u}_{M-1}, \boldsymbol{p}, t_{M-1}] \\ \boldsymbol{\zeta}_{M-1} \\ \boldsymbol{g}[\boldsymbol{y}_M, \boldsymbol{u}_M, \boldsymbol{p}, t_M] \\ \boldsymbol{\psi}_f \end{bmatrix} \tag{102}$$

The resulting formulation is a transcription of the original trajectory optimization problem defined by Eqs. (2–7) into an NLP problem as given by Eqs. (27–29).

Collecting results yields a function generator for the direct transcription method of the following form:

Direct Transcription

Input: $\boldsymbol{x}$
Do for (each phase) $k = 1, N$
  Initialize phase k: save $\boldsymbol{\psi}_0$
  Do for (each grid point) $j = 0, M-1$
    Constraint evaluation: save $\boldsymbol{g}[\boldsymbol{y}_j, \boldsymbol{u}_j, \boldsymbol{p}, t_j]$ and $\boldsymbol{\zeta}_j$
  End do
  Terminate phase
    Save $\boldsymbol{g}[\boldsymbol{y}_M, \boldsymbol{u}_M, \boldsymbol{p}, t_M]$ and $\boldsymbol{\psi}_f$
End do
Terminate trajectory
Output: $\boldsymbol{c}(\boldsymbol{x})$ and $F(\boldsymbol{x})$

The nonlinear programming problem that results from this formulation is large. It is clear that the number of NLP variables $n \approx (n_y + n_u)MN$, with a similar number of NLP constraints. Thus a typical trajectory with 7 states, 2 controls, 100 grid points per phase, and 5 phases produces an NLP with $n = 4500$. Fortunately, the pertinent matrices for this NLP problem, namely, the Jacobian $\boldsymbol{G}$ and the Hessian $\boldsymbol{H}$, are also sparse. So for an application of this type it would not be unusual for these matrices to have fewer than 1% of the elements be nonzero. Consequently, exploiting sparsity to reduce both storage and computation time is a critical aspect of a successful implementation when using a direct transcription method.

2. Examples

The optimal trajectories by implicit simulation (OTIS) program originally proposed by Hargraves and Paris[84] implements the basic collocation method, in addition to a more standard direct shooting approach. The original implementation has been widely distributed to NASA, the U.S. Air Force, and academic and commercial institutions throughout the United States. Early versions of the tool utilized the Hermite–Simpson defect[79] and the NPSOL nonlinear programming software.[7] More recent versions of the OTIS software have incorporated the sparse optimal control software (SOCS).[85–89] The OTIS/SOCS library incorporates a number of features not available in earlier versions. In particular, a sparse nonlinear programming algorithm[11,12,14,15] permits the solution of problems with $n \approx m \approx 100{,}000$ on an engineering workstation. The sparse NLP implements a sparse SQP method based on a Schur-complement algorithm suggested by Gill et al.[6,16] Jacobian and Hessian matrices are computed efficiently using sparse finite differencing as proposed by Coleman and Moré[18] and Curtis et al.[19] Automatic refinement of the mesh to achieve specified accuracy in the discretization is available.[90] Like other widely used production tools for trajectory design, OTIS provides an extensive library of models to permit ascent, re-entry, and orbital simulations. The SOCS software has also been used independent of the OTIS software for low-thrust orbit transfers,[91] low-thrust interplanetary transfers,[92] commercial aircraft mission analysis,[93] and applications in chemical process control and robotics. The direct transcription method has also been implemented by Enright and Conway.[34,94] The advanced launch trajectory optimization software (ALTOS) program,[95,96] developed in Germany for the European Space Agency, incorporates many of the same features. A sparse reduced gradient method has been investigated for collocation problems by Brenan and Hallman,[97] and Steinbach has investigated an interior point SQP approach.[98]

3. Issues

When the direct transcription method is implemented using a sparse nonlinear programming algorithm, the overall approach does resolve many of the difficulties encountered in trajectory optimization. Some limitations can be attributed to the underlying sparse NLP algorithms. For example, one of the principal attractions of a direct method is that adjoint equations do not need to be computed. On the other hand, the underlying NLP must in fact use derivative information that is in some sense equivalent. The sparse NLP used in SOCS computes Jacobian and Hessian information by sparse finite differencing. This technique will be efficient as long as the number of perturbations is reasonably small. The number of perturbations are determined by the DAE right-hand-side matrices $\boldsymbol{f}_y, \boldsymbol{f}_u, \boldsymbol{g}_y$, and $\boldsymbol{g}_u$. If these matrices are dense, then the number of perturbations γ needed by the sparse differencing technique is approximately equal to the number of state and control variables, i.e., $\gamma \approx n_y + n_u$. Hessian information can be computed using $\gamma^2/2$ perturbations. Thus as long as γ is small, this technique is reasonable. Conversely, the sparse differencing approach is too expensive when the size of the DAE system becomes large. On the other hand, when the right-hand-side matrices $\boldsymbol{f}_y, \boldsymbol{f}_u, \boldsymbol{g}_y$, and $\boldsymbol{g}_u$ are sparse, then it may be that $\gamma \ll n_y + n_u$. Stated simply, sparse finite differencing must either exploit right-hand-side sparsity or is limited by the number of state and control variables. Reduced gradient algorithms[97] and reduced SQP methods[20] are limited in a different way. Algorithms of this type construct the reduced Hessian, which is a dense $n_d \times n_d$ matrix where the number of degrees of freedom is approximately equal to the number of controls times the number of grid points, i.e., $n_d \approx n_u M$. Because the reduced Hessian is dense and linear algebra operations are $\mathcal{O}(n_d^3)$, this implies an upper limit with $n_u M \approx 500$. Consequently, NLP methods that form the reduced Hessian are limited by the number of control variables and/or the number of mesh points.

The second principle attraction involves the treatment of path inequality constraints. Defining an a priori distribution of constrained subarcs is not necessary because the underlying NLP effectively determines the subarcs using an active set strategy. However, this approach is really only an approximation to the true solution. First, the number of grid points effectively defines the resolution of the constrained subarcs. Second, when entering or leaving a path constraint the control time history may require jump discontinuities at the junction points. If the control approximation does not permit discontinuities, then the result will be suboptimal. One can of course introduce a phase boundary at the junction point that will improve the control approximation. However, this technique has not been automated.

A third more fundamental difficulty occurs when the underlying DAE has an index greater than 1, as with singular arcs and/or state variable inequalities. In this case, either the NLP subproblem is singular and/or attempts to refine the mesh for improved accuracy will fail. Essentially these difficulties can be attributed to the fact that none of the discretization schemes described are appropriate for high index DAEs. One approach is to impose additional nonlinear path conditions along the singular arc.[99] Unfortunately, this approach requires analytic elimination of the adjoint variables and a priori knowledge of the constrained subarcs. A more promising approach is to incorporate a four-point collocation scheme during the singular arc as proposed by Logsdon and Biegler.[100]

VII. Other Methods

The vast majority of successful trajectory optimization applications incorporate some variant of the methods described earlier. For the sake of completeness, we include a brief discussion of two other approaches that have been considered but that are generally not computationally competitive.

A. Dynamic Programming

In the late 1950s Bellman[101] introduced a generalization of the classical Hamilton–Jacobi theory. The key notion of these so-called extremal field methods is described by a system of first-order nonlinear partial differential equations known as the Hamilton–Jacobi–Bellman equation. Essentially, these PDEs describe the optimal control functions $\boldsymbol{u}^*[\boldsymbol{x}, t]$ as well as the optimal objective for all possible initial conditions $[\boldsymbol{x}(t_0), t_0]$. Hamilton–Jacobi–Bellman theory has played a major role in the development of necessary and sufficient conditions and has provided a unified theoretical basis for the field of optimal control. Despite its theoretical importance, the utility of dynamic programming as the basis for a viable numerical method is summarized by Bryson and Ho on page 136 of Ref. 23 as follows:

> The great drawback of dynamic programming is, as Bellman himself calls it, the "curse of dimensionality." Even recording the solution to a moderately complicated problem involves an enormous amount of storage. If we want only one optimal path from a known initial point, it is wasteful and tedious, to find a whole field of extremals. . . .

B. Genetic Algorithms

All of the trajectory optimization methods described earlier have well-defined termination criteria. As a consequence, it is possible to decide whether a candidate solution, e.g., $\hat{\boldsymbol{x}}$, is in fact an answer by evaluating the necessary conditions, e.g., Eqs. (20) and (21) or (36–40). The ability to define convergence is a fundamental property of calculus-based methods. In contrast, when the variables are discrete, calculus-based methods do not apply. In general, for problems with discrete variables the only way to decide if a candidate solution $\hat{\boldsymbol{x}}$ is in fact an answer is by comparison with all other possible candidates. Unfortunately, this is a combinatorial problem that is computationally prohibitive for all but the smallest applications. To avoid direct comparison of all possible solutions, it is necessary to introduce randomness at some point in the optimization process and abandon a definitive convergence criterion. The basic notion of genetic algorithms, simulated annealing, tabu search, and evolutionary or Monte Carlo methods is to randomly select values for the unknown problem variables. After a finite number of random samples the best value is considered the answer. For some applications, notably those with discrete variables, algorithms of this type are the only practical alternative. However, trajectory optimization problems do not fall in this class! Trajectory applications are not characterized by discrete variables, and there simply is no reason to use a method that incurs the penalty associated with this assumption. Nevertheless, methods of this type have attracted the interest of many analysts, presumably because they are incredibly simple to apply without a detailed understanding of the system being optimized. Unfortunately, because they do not exploit gradient information, they are not computationally competitive with the methods in Sec. VI.

VIII. Conclusions

There are many techniques for numerically solving trajectory optimization problems, and it is sometimes helpful to classify the techniques as either indirect or direct. Indirect methods are characterized by explicitly solving the optimality conditions stated in terms of the adjoint differential equations, the maximum principle, and associated boundary (transversality) conditions. Using the calculus of variations, the optimal control necessary conditions can be derived by setting the first variation of the Hamiltonian function to zero. The indirect approach usually requires the solution of a nonlinear multipoint boundary value problem. By analogy, an indirect method for optimizing a function of n variables would require analytically computing the gradient and then locating a set of variables using a root-finding algorithm such that the gradient is zero. There are three primary drawbacks to this approach in practice. First, it is necessary to derive analytic expressions for the necessary conditions, and for complicated nonlinear dynamics this can become quite daunting. Second, the region of convergence for a root-finding algorithm may be surprisingly small, especially when it is necessary to guess values for the adjoint variables that may not have an obvious physical interpretation. Third, for problems with path inequalities it is necessary to guess the sequence of constrained and unconstrained subarcs before iteration can begin.

In contrast, a direct method does not require an analytic expression for the necessary conditions and typically does not require initial guesses for the adjoint variables. Instead, the dynamic (state and control) variables are adjusted to directly optimize the objective function. All direct methods introduce some parametric representation for the control variables. For simple shooting, the control functions are defined by a relatively small number of NLP variables. For multiple shooting and transcription methods, the number of NLP variables used to describe the controls increases, ultimately including values at each mesh point in the interval.

Throughout the paper I have emphasized the similarity between methods. All of the methods utilize a Newton-based iteration to adjust a finite set of variables. The methods can be distinguished by identifying the set of iteration variables and constraints. The optimal control necessary conditions can be interpreted as limiting forms of the NLP Kuhn–Tucker necessary conditions. At the present time perhaps the most widely used methods are direct shooting, indirect multiple shooting, and direct transcription. Each method has advantages and disadvantages, and I have attempted to highlight them. Future research and development will undoubtedly focus on removing deficiencies in these techniques. Progress in the analysis of high-index differential-algebraic equations, automatic differentiation, and sparse nonlinear programming will certainly lead to refinements in existing software and methods. In fact, one may expect many of the best features of seemingly disparate techniques to merge, forming still more powerful methods.

References

[1] Hallman, W. P., “Mission Timeline for a Shuttle-IUS Flight Using a Nonstandard Shuttle Park Orbit,” The Aerospace Corp., Technical Operating Rept. TOR-0083-(3493-14), El Segundo, CA, Oct. 1982.

[2] Citron, S. J., *Elements of Optimal Control*, Holt, Rinehart and Winston, New York, 1969.

[3] Fletcher, R., *Practical Methods of Optimization, Vol. 2, Constrained Optimization*, Wiley, New York, 1985.

[4] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic, London, 1981.

[5] Dennis, J. E., Jr., and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[6] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., “Some Theoretical Properties of an Augmented Lagrangian Merit Function,” Dept. of Operations Research, Stanford Univ., TR SOL 86-6, Stanford, CA, April 1986.

[7] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., “User’s Guide for NPSOL (version 4.0): A Fortran Package for Nonlinear Programming,” Dept. of Operations Research, Stanford Univ., TR SOL 86-2, Stanford, CA, Jan. 1986.

[8] Betts, J. T., and Hallman, W. P., “NLP2 Optimization Algorithm Documentation,” The Aerospace Corp., Technical Operating Rept. TOR-0089(4464-06)-1, reissue A, El Segundo, CA, 1997.

[9] Rosen, J. B., and Kreuser, J., “A Gradient Projection Algorithm for Nonlinear Constraints,” *Numerical Methods for Non-Linear Optimization*, edited by F. A. Lootsma, Academic, London, 1972, pp. 297–300.

[10] Lasdon, L. S., and Waren, A. D., “Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Optimization,” *Design and Implementation of Optimization Software*, edited by H. J. Greenberg, Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 1978, pp. 335–362.

[11] Betts, J. T., and Frank, P. D., “A Sparse Nonlinear Optimization Algorithm,” *Journal of Optimization Theory and Applications*, Vol. 82, 1994, pp. 519–541.

[12] Betts, J. T., Carter, M. J., and Huffman, W. P., “Software for Nonlinear Optimization,” Boeing Information and Support Services, Mathematics and Engineering Analysis Library Rept. MEA-LR-83 R1, The Boeing Co., Seattle, WA, June 1997.

[13] Ashcraft, C. C., Grimes, R. G., and Lewis, J. G., “Accurate Symmetric Indefinite Linear Equation Solvers,” *SIAM Journal of Matrix Analysis* (submitted for publication).

[14] Betts, J. T., Eldersveld, S. K., and Huffman, W. P., “A Performance Comparison of Nonlinear Programming Algorithms for Large Sparse Problems,” *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1993 (AIAA Paper 93-3751).

[15] Betts, J. T., and Huffman, W. P., “Sparse Nonlinear Programming Test Problems (release 1.0),” Boeing Computer Services, BCS Technology TR BCSTECH-93-047, The Boeing Co., Seattle, WA, 1993.

[16] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., “A Schur-Complement Method for Sparse Quadratic Programming,” Dept. of Operations Research, Stanford Univ., TR SOL 87-12, Stanford, CA, Oct. 1987.

[17] Betts, J. T., “Sparse Jacobian Updates in the Collocation Method for Optimal Control Problems,” *Journal of Guidance, Control, and Dynamics*, Vol. 13, 1990, pp. 409–415.

[18] Coleman, T. F., and Moré, J. J., “Estimation of Sparse Jacobian Matrices and Graph Coloring Problems,” *SIAM Journal of Numerical Analysis*, Vol. 20, 1983, pp. 187–209.

[19] Curtis, A. R., Powell, M., and Reid, J., “On the Estimation of Sparse Jacobian Matrices,” *Journal of the Institute of Mathematics and Applications*, Vol. 13, 1974, pp. 117–120.

[20] Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” Dept. of Operations Research, Stanford Univ., TR SOL 96-0, Stanford, CA, 1996.

[21] Bliss, G. A., *Lectures on the Calculus of Variations*, Univ. of Chicago Press, Chicago, IL, 1946.

[22] Pontryagin, L. S., *The Mathematical Theory of Optimal Processes*, Wiley-Interscience, New York, 1962.

[23] Bryson, A. E., Jr., and Ho, Y. C., *Applied Optimal Control*, Wiley, New York, 1975.

[24]Ascher, U. M., Mattheij, R. M. M., and Russell, R. D., *Numerical Solution of Boundary Value Problems in Ordinary Differential Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1988.

[25]Brenan, K. E., Campbell, S. L., and Petzold, L. R., *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, New York, 1989.

[26]Tsien, H. S., and Evans, R. C., "Optimum Thrust Programming for a Sounding Rocket," *Journal of the American Rocket Society*, Vol. 21, 1951, pp. 99–107.

[27]Speyer, J. L., "Periodic Optimal Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 745–755.

[28]Sachs, G., and Lesch, K., "Periodic Maximum Range Cruise with Singular Control," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 790–793.

[29]Berkmann, P., and Pesch, H. J., "Abort Landing in Windshear: Optimal Control Problem with Third-Order State Constraint and Varied Switching Structure," *Journal of Optimization Theory and Applications*, Vol. 85, 1995, pp. 21–57.

[30]Pesch, H. J., "A Practical Guide to the Solution of Real-Life Optimal Control Problems," *Control and Cybernetics*, Vol. 23, 1994, pp. 7–60.

[31]Canon, M. D., Cullum, C. D., and Polak, E., *Theory of Optimal Control and Mathematical Programming*, McGraw–Hill, New York, 1970.

[32]Polak, E., *Computational Methods in Optimization*, Academic, New York, 1971.

[33]Tabak, D., and Kuo, B. C., *Optimal Control by Mathematical Programming*, Prentice–Hall, Englewood Cliffs, NJ, 1971.

[34]Enright, P. J., and Conway, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, 1992, pp. 994–1002.

[35]Herman, A. L., and Conway, B. A., "Direct Optimization Using Collocation Based on High-Order Gauss–Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.

[36]Von Stryk, O., and Bulirsch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, 1992, pp. 357–373.

[37]Von Stryk, O., "Numerical Solution of Optimal Control Problems by Direct Collocation," *Optimal Control*, edited by R. Bulirsch, A. Miele, J. Stoer, and K. H. Wells, Vol. 111, International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, Switzerland, 1993, pp. 129–143.

[38]Calise, A. J., "Singular Perturbation Techniques for On-Line Optimal Flight-Path Control," *Journal of Guidance and Control*, Vol. 4, 1981, pp. 398–405.

[39]Calise, A. J., and Moerder, D. D., "Singular Perturbation Techniques for Real Time Aircraft Trajectory Optimization and Control," NASA CR-3597, Aug. 1982.

[40]Bashforth, F., and Adams, J. C., *Theories of Capillary Action*, Cambridge Univ. Press, New York, 1883.

[41]Moulton, F. R., *New Methods in Exterior Ballistics*, Univ. of Chicago, Chicago, IL, 1926.

[42]Dahlquist, G., and Björk, Å., *Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1974.

[43]Stoer, J., and Bulirsch, R., *Introduction to Numerical Analysis*, Springer, New York, 1980.

[44]Hindmarsh, A. C., "ODEPACK, A Systematized Collection of ODE Solvers," *Scientific Computing*, edited by R. S. Stepleman, North-Holland, Amsterdam, 1983, pp. 55–64.

[45]Shampine, L. F., and Gordon, M. K., *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. H. Freeman and Co., San Francisco, 1975.

[46]Gear, C. W., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1971.

[47]Seywald, H., "Trajectory Optimization Based on Differential Inclusion," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1994, pp. 480–487.

[48]Gear, C. W., "The Simultaneous Numerical Solution of Differential-Algebraic Equations," *IEEE Transactions on Circuit Theory*, Vol. CT-18, 1971, pp. 89–95.

[49]Petzold, L. R., "Differential/Algebraic Equations Are Not ODEs," *SIAM Journal on Scientific and Statistical Computing*, Vol. 3, 1982, pp. 367–384.

[50]Petzold, L. R., "A Description of DASSL, A Differential/Algebraic System Solver," *Scientific Computing*, edited by R. S. Stepleman, North-Holland, Amsterdam, 1983, pp. 65–68.

[51]Hallman, W. P., "Smooth Curve Fits for the Shuttle Solid Rocket Booster Data," The Aerospace Corp., Interoffice Correspondence IOC A85-5752.5-05, El Segundo, CA, March 1985.

[52]Luke, R. A., "Computational Efficiency Considerations for High-Fidelity Launch Vehicle Trajectory Optimization," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1989 (AIAA Paper 89-3446).

[53]De Boor, C., *A Practical Guide to Splines*, Springer–Verlag, New York, 1978.

[54]Ferguson, D. R., and Mastro, R. A., "Modeling and Analysis of Aerodynamic Data II. Practical Experience," *Proceedings of the AIAA/AHS/ASEE Aircraft Design, Systems and Operations Conference*, AIAA, Washington, DC, 1989 (AIAA Paper 89-2076).

[55]Ferguson, D. R., and Mastro, R. A., "Modeling and Analysis of Aerodynamic Data II," AIAA Paper 89-0476, June 1989.

[56]Ferguson, D. R., "Construction of Curves and Surfaces Using Numerical Optimization Techniques," *CAD*, Vol. 18, 1986, pp. 15–21.

[57]Betts, J. T., "The Application of Sparse Least Squares in Aerospace Design Problems," *Optimal Design and Control, Proceedings of the Workshop on Optimal Design and Control*, edited by J. Borggaard, J. Burkardt, M. Gunzburger, and J. Peterson, Vol. 19, Progress in Systems and Control Theory, Birkhäuser, Basel, Switzerland, 1994, pp. 81–96.

[58]Brauer, G. L., Cornick, D. E., and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," NASA CR-2770, Feb. 1977.

[59]Meder, D. S., and Searcy, J. L., "Generalized Trajectory Simulation (GTS), Volumes I-V," The Aerospace Corp., TR SAMSO-TR-75-255, El Segundo, CA, Jan. 1975.

[60]Betts, J. T., "Optimal Three-Burn Orbit Transfer," *AIAA Journal*, Vol. 15, 1977, pp. 861–864.

[61]Brusch, R. G., "Constrained Impulsive Trajectory Optimization for Orbit-to-Orbit Transfer," *Journal of Guidance and Control*, Vol. 2, 1979, pp. 204–212.

[62]Bauer, T. P., Betts, J. T., Hallman, W. P., Huffman, W. P., and Zondervan, K. P., "Solving the Optimal Control Problem Using a Nonlinear Programming Technique Part 2: Optimal Shuttle Ascent Trajectories," *Proceedings of the AIAA/AAS Astrodynamics Conference*, AIAA, Washington, DC, 1984 (AIAA Paper 84-2038).

[63]Brenan, K. E., "Engineering Methods Report: The Design and Development of a Consistent Integrator/Interpolator for Optimization Problems," The Aerospace Corp., ATM 88(9975)-52, El Segundo, CA, 1988.

[64]Gear, C. W., and Van Vu, T., "Smooth Numerical Solutions of Ordinary Differential Equations," *Proceedings of the Workshop on Numerical Treatment of Inverse Problems for Differential and Integral Equations*, Heidelberg, Germany, 1982.

[65]Van Vu, T., "Numerical Methods for Smooth Solutions of Ordinary Differential Equations," Dept. of Computer Science, Univ. of Illinois, TR UIUCDCS-R-83-1130, Urbana, IL, May 1983.

[66]Betts, J. T., "Frontiers in Engineering Optimization," *Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 105, 1983, pp. 151–154.

[67]Balkanyi, L. R., and Spurlock, O. F., "DUKSUP—A High Thrust Trajectory Optimization Code," *Proceedings of the AIAA/AHS/ASEE Aerospace Design Conference*, AIAA, Washington, DC, 1993 (AIAA Paper 93-1127).

[68]Edelbaum, T., Sackett, L., and Malchow, H., "Optimal Low Thrust Geocentric Transfer," *Proceedings of the AIAA 10th Electric Propulsion Conference*, AIAA, New York, 1973 (AIAA Paper 73-1074).

[69]Bischof, C., Carle, A., Corliss, G., Griewank, A., and Hovland, P., "ADIFOR: Generating Derivative Codes from Fortran Programs," *Scientific Programming*, Vol. 1, 1992, pp. 11–29.

[70]Schöpf, R., and Deulfhard, P., "OCCAL: A Mixed Symbolic-Numeric Optimal Control CALculator," *Control Applications of Optimization*, edited by R. Bulirsch and Dieter Kraft, Birkhäuser Verlag, Basel, Switzerland, 1993.

[71]Mehlhorn, R., and Sachs, G., "A New Tool for Efficient Optimization by Automatic Differentiation and Program Transparency," *Optimization Methods and Software*, Vol. 4, 1994, pp. 225–242.

[72]Bulirsch, R., "Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung," Rept. of the Carl-Cranz Gesellschaft, Carl-Cranz Gesellschaft, Oberpfaffenhofen, Germany, 1971.

[73]Keller, H. B., *Numerical Methods for Two-Point Boundary Value Problems*, Blaisdell, London, 1968.

[74]Oberle, H. J., and Grimm, W., "BNDSCO—A Program for the Numerical Solution of Optimal Control Problems, User Guide," Institut für Dynamik der Flugsysteme, Deutsche Forschungsanstalt für Luft und Raumfahrt DLR, Internal Rept. DLR IB/515-89/22, Oberpfaffenhofen, Germany, 1989.

[75]Callies, R., "Optimal Design of a Mission to Neptune," *Optimal Control*, edited by R. Bulirsch, A. Miele, J. Stoer, and K. H. Wells, Vol. 111, International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, Switzerland, 1993, pp. 341–349.

[76]Kreim, H., Kugelmann, B., Pesch, H. J., and Breitner, M. H., "Minimizing the Maximum Heating of a Re-Entering Space Shuttle: An Optimal Control Problem with Multiple Control Constraints," *Optimal Control Applications and Methods*, Vol. 17, 1996, pp. 45–69.

[77]Bock, H. G., and Plitt, K. J., "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems," *Proceedings of the 9th IFAC World Congress*, Pergamon, 1984, pp. 242–247.

[78]Betts, J. T., and Huffman, W. P., "Trajectory Optimization on a Parallel Processor," *Journal of Guidance, Control, and Dynamics*, Vol. 14, 1991, pp. 431–439.

[79]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*, Vol. IFIP-TC7, Springer–Verlag, New York, 1975.

[80]Russell, R. D., and Shampine, L. F., "A Collocation Method for Boundary Value Problems," *Numerische Mathematik*, Vol. 19, 1972, pp. 13–36.

[81]Dickmanns, E. D., "Efficient Convergence and Mesh Refinement Strategies for Solving General Ordinary Two-Point Boundary Value Problems by Collocated Hermite Approximation," 2nd IFAC Workshop on Optimisation, Oberpfaffenhofen, Germany, Sept. 1980.

[82]Ascher, U. M., Christiansen, J., and Russell, R. D., "COLSYS—A Collocation Code for Boundary-Value Problems," *Codes for Boundary Value Problems in Ordinary Differential Equations*, edited by B. Childs, M. Scott, J. W. Daniel, E. Denman, and P. Nelson, Vol. 76, Lecture Notes in Computer Science, Springer–Verlag, Berlin, 1979.

[83]Hull, D. G., "Conversion of Optimal Control Problems into Parameter Optimization Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 57–60.

[84]Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, 1987, p. 338.

[85]Betts, J. T., and Huffman, W. P., "Application of Sparse Nonlinear Programming to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, 1992, pp. 198–206.

[86]Betts, J. T., and Huffman, W. P., "Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 1, 1993, pp. 59–68.

[87]Betts, J. T., "Issues in the Direct Transcription of Optimal Control Problems to Sparse Nonlinear Programs," *Computational Optimal Control*, edited by R. Bulirsch and D. Kraft, Vol. 115, International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, Switzerland, 1994, pp. 3–18.

[88]Betts, J. T., "Trajectory Optimization Using Sparse Sequential Quadratic Programming," *Optimal Control*, edited by R. Bulirsch, A. Miele, J. Stoer, and K. H. Wells, Vol. 111, International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, Switzerland, 1993, pp. 115–128.

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

[90]Betts, J. T., and Huffman, W. P., "Mesh Refinement in Direct Transcription Methods for Optimal Control," *Optimal Control Applications and Methods* (to be published).

[91]Betts, J. T., "Using Sparse Nonlinear Programming to Compute Low Thrust Orbit Transfers," *Journal of the Astronautical Sciences*, Vol. 41, 1993, pp. 349–371.

[92]Betts, J. T., "Optimal Interplanetary Orbit Transfers by Direct Transcription," *Journal of the Astronautical Sciences*, Vol. 42, 1994, pp. 247–268.

[93]Betts, J. T., and Cramer, E. J., "Application of Direct Transcription to Commercial Aircraft Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 151–159.

[94]Enright, P. J., and Conway, B. A., "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 14, 1991, pp. 981–985.

[95]Roenneke, A. J., Jänsch, C., and Markl, A., "Advanced Launch and Reentry Trajectory Optimization Software Documentation (ALTOS)," European Space Science and Technology Center, TR, Software User Manual, Draft 1, Noordwijk, The Netherlands, May 1995.

[96]Kraft, D., "On Converting Optimal Control Problems into Nonlinear Programming Problems," *Computational Mathematical Programming*, Vol. F15, NATO ASI Series, Springer–Verlag, 1985.

[97]Brenan, K. E., and Hallman, W. P., "A Generalized Reduced Gradient Algorithm for Large-Scale Trajectory Optimization Problems," *Optimal Design and Control*, Proceedings of the Workshop on Optimal Design and Control, edited by J. Borggaard, J. Burkardt, M. Gunzburger, and J. Peterson, Vol. 19, Progress in Systems and Control Theory, Birkhäuser Verlag, Basel, Switzerland, 1994, pp. 117–132.

[98]Steinbach, M. C., "A Structured Interior Point SQP Method for Nonlinear Optimal Control Problems," *Computational Optimal Control*, edited by R. Bulirsch and D. Kraft, Vol. 115, International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, Switzerland, 1994, pp. 213–222.

[99]Downey, J. R., and Conway, B. A., "The Solution of Singular Optimal Control Problems Using Direct Collocation and Nonlinear Programming," AAS/AIAA Astrodynamics Specialist Conf., AAS Paper 91-443, Durango, CO, Aug. 1991.

[100]Logsdon, J. S., and Biegler, L. T., "Accurate Solution of Differential-Algebraic Optimization Problems," *Industrial Engineering Chemical Research*, Vol. 28, 1989, pp. 1628–1639.

[101]Bellman, R., *Dynamic Programming*, Princeton Univ. Press, Princeton, NJ, 1957.

This article has been cited by:

1. David Braun, Matthew Howard, Sethu Vijayakumar. 2012. Optimal variable stiffness control: formulation and application to explosive movement tasks. *Autonomous Robots* **33**:3, 237-253. [CrossRef]
2. G. Mingotti, F. Topputo, F. Bernelli-Zazzera. 2012. Transfers to distant periodic orbits around the Moon via their invariant manifolds. *Acta Astronautica* **79**, 20-32. [CrossRef]
3. Ashley MooreJerrold E MarsdenSina Ober-BlöbaumShane RossJon A SimsMarin KobilarovEvan GawlikNatasha Bosanac. 2012. Trajectory Design Combining Invariant Manifolds with Discrete Mechanics and Optimal Control. *Journal of Guidance, Control, and Dynamics* **35**:5, 1507-1525. [Citation] [PDF] [PDF Plus]
4. Michael McAsey, Libin Mou, Weimin Han. 2012. Convergence of the forward-backward sweep method in optimal control. *Computational Optimization and Applications* **53**:1, 207-226. [CrossRef]
5. E. Trélat. 2012. Optimal Control and Applications to Aerospace: Some Results and Challenges. *Journal of Optimization Theory and Applications* **154**:3, 713-758. [CrossRef]
6. GuoQiang Huang, YuPing Lu, Ying Nan. 2012. A survey of numerical algorithms for trajectory optimization of flight vehicles. *Science China Technological Sciences* **55**:9, 2538-2560. [CrossRef]
7. Luitpold Babel. 2012. Three-dimensional Route Planning for Unmanned Aerial Vehicles in a Risk Environment. *Journal of Intelligent & Robotic Systems* . [CrossRef]
8. Kathrin Flaßkamp, Sina Ober-Blöbaum, Marin Kobilarov. 2012. Solving Optimal Control Problems by Exploiting Inherent Dynamical Systems Structures. *Journal of Nonlinear Science* **22**:4, 599-629. [CrossRef]
9. F. FischJ. LenzF. HolzapfelG. Sachs. 2012. On the Solution of Bilevel Optimal Control Problems to Increase the Fairness in Air Races. *Journal of Guidance, Control, and Dynamics* **35**:4, 1292-1298. [Citation] [PDF] [PDF Plus]
10. Mauro PontaniPradipto GhoshBruce A. Conway. 2012. Particle Swarm Optimization of Multiple-Burn Rendezvous Trajectories. *Journal of Guidance, Control, and Dynamics* **35**:4, 1192-1207. [Citation] [PDF] [PDF Plus]
11. Ossama AbdelkhalikEhsan Taheri. 2012. Shape Based Approximation of Constrained Low-Thrust Space Trajectories using Fourier Series. *Journal of Spacecraft and Rockets* **49**:3, 535-546. [Citation] [PDF] [PDF Plus]
12. Mathieu Balesdent, Nicolas Bérend, Philippe Dépincé, Abdelhamid Chriette. 2012. A survey of multidisciplinary design optimization methods in launch vehicle design. *Structural and Multidisciplinary Optimization* **45**:5, 619-642. [CrossRef]
13. Chris T. Freeman. 2012. Constrained point-to-point iterative learning control with experimental verification. *Control Engineering Practice* **20**:5, 489-498. [CrossRef]
14. Ehsan Taheri, Ossama Abdelkhalik. 2012. Shape-Based Approximation of Constrained Low-Thrust Space Trajectories Using Fourier Series. *Journal of Spacecraft and Rockets* **49**:3, 535-545. [CrossRef]
15. Timothy Jorris, ; Eric Paulson, ; Ryan Carr, Multidisciplinary Design Optimization for a Reusable Launch Vehicle Using Multiple-Phase Pseudospectral Optimization . [Citation] [PDF] [PDF Plus]
16. Huifeng LI, Ran ZHANG, Zhaoying LI, Rui ZHANG. 2012. Footprint Problem with Angle of Attack Optimization for High Lifting Reentry Vehicle. *Chinese Journal of Aeronautics* **25**:2, 243-251. [CrossRef]
17. J.P. Sanchez, C.R. McInnes. 2012. Assessment on the feasibility of future shepherding of asteroid resources. *Acta Astronautica* **73**, 49-66. [CrossRef]
18. Marko Ackermann, Antonie J. van den Bogert. 2012. Predictive simulation of gait at low gravity reveals skipping as the preferred locomotion strategy. *Journal of Biomechanics* **45**:7, 1293-1298. [CrossRef]
19. Karl J ObermeyerPaul OberlinSwaroop DarbhaMark MearsSteven SmithVitaly ShafermanDerek KingstonMaruthi Akella. 2012. Sampling-Based Path Planning for a Visual Reconnaissance Unmanned Air Vehicle. *Journal of Guidance, Control, and Dynamics* **35**:2, 619-631. [Citation] [PDF] [PDF Plus]
20. Haijun PengQiang GaoZhigang WuWanxie ZhongMehran MesbahiBingen YangDaniel Scheeres. 2012. Symplectic Approaches for Solving Two-Point Boundary-Value Problems. *Journal of Guidance, Control, and Dynamics* **35**:2, 653-659. [Citation] [PDF] [PDF Plus]
21. Jörg Fliege, Konstantinos Kaparis, Banafsheh Khosravi. 2012. Operations research in the space industry. *European Journal of Operational Research* **217**:2, 233-240. [CrossRef]
22. C. Pukdeboon, A.S.I. Zinober. 2012. Control Lyapunov function optimal sliding mode controllers for attitude tracking of spacecraft. *Journal of the Franklin Institute* **349**:2, 456-475. [CrossRef]

23. Federico Zuiani, Massimiliano Vasile, Alessandro Palmas, Giulio Avanzini. 2012. Direct transcription of low-thrust trajectories with finite trajectory elements. *Acta Astronautica* **72**, 108-120. [CrossRef]
24. Luciano Blasi, Simeone Barbato, Massimiliano Mattei. 2012. A particle swarm approach for flight path optimization in a constrained environment. *Aerospace Science and Technology* . [CrossRef]
25. J.P. Sanchez, C.R. McInnes. 2012. Synergistic approach of asteroid exploitation and planetary protection. *Advances in Space Research* **49**:4, 667-685. [CrossRef]
26. John Gregory, Alberto Olivares, Ernesto Staffetti. 2012. Energy-optimal trajectory planning for the Pendubot and the Acrobot. *Optimal Control Applications and Methods* n/a-n/a. [CrossRef]
27. A. Zanzottera, G. Mingotti, R. Castelli, M. Dellnitz. 2012. Intersecting invariant manifolds in spatial restricted three-body problems: Design and optimization of Earth-to-halo transfers in the Sun–Earth–Moon scenario. *Communications in Nonlinear Science and Numerical Simulation* **17**:2, 832-843. [CrossRef]
28. Bruce A. Conway. 2012. A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems. *Journal of Optimization Theory and Applications* **152**:2, 271-306. [CrossRef]
29. G. Mingotti, F. Topputo, F. Bernelli-Zazzera. 2012. Efficient invariant-manifold, low-thrust planar trajectories to the Moon. *Communications in Nonlinear Science and Numerical Simulation* **17**:2, 817-831. [CrossRef]
30. Zewei Zheng, ; Ming Zhu, ; Haoquan Liang, ; Xiao Guo, Ascent Trajectory Optimization for Stratospheric Airships with Thermal Effects . [Citation] [PDF] [PDF Plus]
31. Joaquim Martins, ; John Hwang, A Dynamic Parametrization Scheme for Shape Optimization Using Quasi-Newton Methods . [Citation] [PDF] [PDF Plus]
32. Ronaldo Dias, Nancy L. Garcia, Adriano Z. Zambom. 2012. Monte Carlo algorithm for trajectory optimization based on Markovian readings. *Computational Optimization and Applications* **51**:1, 305-321. [CrossRef]
33. Geoffrey G. Wawrzyniak, Kathleen C. Howell. 2011. Numerical techniques for generating and refining solar sail trajectories. *Advances in Space Research* **48**:11, 1848-1857. [CrossRef]
34. Hoam Chung, Elijah Polak, Johannes O. Royset, Shankar Sastry. 2011. On the optimal detection of an underwater intruder in a channel using unmanned underwater vehicles. *Naval Research Logistics (NRL)* **58**:8, 804-820. [CrossRef]
35. Joris T. OlympioRyan P RusselChristopher Ranieri. 2011. Optimal Control Problem for Low-Thrust Multiple Asteroid Tour Missions. *Journal of Guidance, Control, and Dynamics* **34**:6, 1709-1720. [Citation] [PDF] [PDF Plus]
36. Shinji MitaniHiroshi Yamakawa. 2011. Novel Nonlinear Rendezvous Guidance Scheme Under Constraints on Thrust Direction. *Journal of Guidance, Control, and Dynamics* **34**:6, 1656-1671. [Citation] [PDF] [PDF Plus]
37. Giorgio MingottiFrancesco TopputoFranco Bernelli-ZazzeraEdward BelbrunoMarian GideaGiovanni MengaliAlessandro QuartaChristian Circi. 2011. Optimal Low-Thrust Invariant Manifold Trajectories via Attainable Sets. *Journal of Guidance, Control, and Dynamics* **34**:6, 1644-1656. [Citation] [PDF] [PDF Plus]
38. M. Shamsi. 2011. A modified pseudospectral scheme for accurate solution of Bang-Bang optimal control problems. *Optimal Control Applications and Methods* **32**:6, 668-680. [CrossRef]
39. A. Da Ronch, M. Ghoreyshi, K.J. Badcock. 2011. On the generation of flight dynamics aerodynamic tables by computational fluid dynamics. *Progress in Aerospace Sciences* **47**:8, 597-620. [CrossRef]
40. Bong-Gyun Park, Jong-Sun Ahn, Min-Jea Tahk. 2011. Two-Dimensional Trajectory Optimization for Soft Lunar Landing Considering a Landing Site. *International Journal of Aeronautical and Space Sciences* **12**:3, 288-295. [CrossRef]
41. Di Wu, ; Yiyuan Zhao, Optimization and Sensitivity Analysis of Climb and Descent Trajectories for Reducing Fuel Burn and Emissions . [Citation] [PDF] [PDF Plus]
42. Iman AlizadehBenjamin VillacRyan RussellFirdaus UdwadiaKenneth MeaseBruce ConwayAnastassios Petropoulos. 2011. Static Solutions of the Hamilton-Jacobi-Bellman Equation for Circular Orbit Transfers. *Journal of Guidance, Control, and Dynamics* **34**:5, 1584-1588. [Citation] [PDF] [PDF Plus]
43. John P. W. Stark, Graham G. SwinerdMission Analysis 111-175. [CrossRef]
44. Yi Liu, ; Dingni Zhang, RLV Reentry Trajectory Optimization through Hybridization of an Improved GA and a SQP Algorithm . [Citation] [PDF] [PDF Plus]
45. Elwin de Weerdt, ; Erik-Jan Van Kampen, ; Ping Chu, ; Jan Mulder, Trajectory Optimization Based on Interval Analysis . [Citation] [PDF] [PDF Plus]

46. Florian Holzapfel, ; Florian Fisch, ; Fabian Sewerin, Approach Trajectory Optimization including a Tunnel Track Constraint . [Citation] [PDF] [PDF Plus]

47. James Whidborne, ; Chi Kin Lai, Real-time Trajectory Generation for Collision Avoidance with Obstacle Uncertainty . [Citation] [PDF] [PDF Plus]

48. H. J. Peng, Q. Gao, Z. G. Wu, W. X. Zhong. 2011. Symplectic adaptive algorithm for solving nonlinear two-point boundary value problems in Astrodynamics. *Celestial Mechanics and Dynamical Astronomy* **110**:4, 319-342. [CrossRef]

49. Yihuan LIAO, Daokui LI, Guojin TANG. 2011. Motion Planning for Vibration Reducing of Free-floating Redundant Manipulators Based on Hybrid Optimization Approach. *Chinese Journal of Aeronautics* **24**:4, 533-540. [CrossRef]

50. James Driscoll, ; Sean Torrez, ; Derek Dalle, Multidisciplinary Optimization of the Fuel Consumption of a Dual Mode Scramjet-Ramjet . [Citation] [PDF] [PDF Plus]

51. Reza Jamilnia, Abolghasem Naghash. 2011. Simultaneous optimization of staging and trajectory of launch vehicles using two different approaches. *Aerospace Science and Technology* . [CrossRef]

52. Terumasa Narukawa, Masaki Takahashi, Kazuo Yoshida. 2011. Efficient walking with optimization for a planar biped walker with a torso by hip actuators and springs. *Robotica* **29**:04, 641-648. [CrossRef]

53. Paul Williams. 2011. Quadrature discretization method in tethered satellite control. *Applied Mathematics and Computation* **217**:21, 8223-8235. [CrossRef]

54. G. Mingotti, F. Topputo, F. Bernelli-Zazzera. 2011. Earth–Mars transfers with ballistic escape and low-thrust capture. *Celestial Mechanics and Dynamical Astronomy* **110**:2, 169-188. [CrossRef]

55. Rajnish SharmaSrinivas R. VadaliJohn E. HurtadoP K MenonE Cristiani. 2011. Optimal Nonlinear Feedback Control Design Using a Waypoint Method. *Journal of Guidance, Control, and Dynamics* **34**:3, 698-705. [Citation] [PDF] [PDF Plus]

56. Chris T. Freeman, Zhonglun Cai, Eric Rogers, Paul L. Lewin. 2011. Iterative Learning Control for Multiple Point-to-Point Tracking Application. *IEEE Transactions on Control Systems Technology* **19**:3, 590-600. [CrossRef]

57. Sina Ober-Blöbaum, Oliver Junge, Jerrold E. Marsden. 2011. Discrete mechanics and optimal control: An analysis. *ESAIM: Control, Optimisation and Calculus of Variations* **17**:2, 322-352. [CrossRef]

58. F. Imado. 2011. Pursuit-evasion problems of two cars in an ellipsoid under gravity. *Engineering Optimization* **43**:4, 447-465. [CrossRef]

59. Fumiaki ImadoTakeshi KurodaPing LuMinjea TahkMichael BreitnerRavio Tuomas. 2011. Family of Local Solutions in a Missile-Aircraft Differential Game. *Journal of Guidance, Control, and Dynamics* **34**:2, 583-591. [Citation] [PDF] [PDF Plus]

60. Richard EpenoyColin McInnesMehran MesbahiRandy BeardNicolas PetitDaniel J. Scheeres. 2011. Fuel Optimization for Continuous-Thrust Orbital Rendezvous with Collision Avoidance Constraint. *Journal of Guidance, Control, and Dynamics* **34**:2, 493-503. [Citation] [PDF] [PDF Plus]

61. A. Nikoobin, M. Moradi. 2011. Optimal balancing of robot manipulators in point-to-point motion. *Robotica* **29**:02, 233-244. [CrossRef]

62. Behçet Aç#kme#e, Lars Blackmore. 2011. Lossless convexification of a class of optimal control problems with non-convex control constraints. *Automatica* **47**:2, 341-347. [CrossRef]

63. Alessandro A. Quarta, Giovanni Mengali. 2011. Analytical results for solar sail optimal missions with modulated radial thrust. *Celestial Mechanics and Dynamical Astronomy* **109**:2, 147-166. [CrossRef]

64. André Berger, Alexander Grigoriev, Joyce van Loon. 2011. Price strategy implementation. *Computers & Operations Research* **38**:2, 420-426. [CrossRef]

65. Xavier PratsVicenç PuigJoseba Quevedo. 2011. Equitable Aircraft Noise-Abatement Departure Procedures. *Journal of Guidance, Control, and Dynamics* **34**:1, 192-203. [Citation] [PDF] [PDF Plus]

66. Yiyuan J. ZhaoWei GuoBrian Capozzi. 2011. Optimal Unmanned Aerial Vehicle Flights for Seeability and Endurance in Winds. *Journal of Aircraft* **48**:1, 305-314. [Citation] [PDF] [PDF Plus]

67. D. M NovakM. VasileBradley J WallGiulio AvanziniColin McInnesBruce ConwayPaolo De Pascale. 2011. Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories. *Journal of Guidance, Control, and Dynamics* **34**:1, 128-147. [Citation] [PDF] [PDF Plus]

68. Rushen B PatelPaul J Goulart. 2011. Trajectory Generation for Aircraft Avoidance Maneuvers Using Online Optimization. *Journal of Guidance, Control, and Dynamics* **34**:1, 218-230. [Citation] [PDF] [PDF Plus]

69. Yang Dang-guo, Zhang Zheng-yu, Sun Yan, Zhu Wei-jun. 2011. A preliminary design and manufacturing study of hybrid lightweight high-speed wind-tunnel models. *Rapid Prototyping Journal* **17**:1, 45-54. [CrossRef]

70. Andrea Minelli, Francesco Topputo, Franco Bernelli-Zazzera. 2011. Controlled Drug Delivery in Cancer Immunotherapy: Stability, Optimization, and Monte Carlo Analysis. *SIAM Journal on Applied Mathematics* **71**:6, 2229. [CrossRef]

71. Xiangyuan Zeng, Junfeng Li, Hexi Baoyin, Shengping Gong. 2011. Trajectory optimization and applications using high performance solar sails. *Theoretical and Applied Mechanics Letters* **1**:3, 033001. [CrossRef]

72. D. H. A. Maithripala, D. H. S. Maithripala, S. Jayasuriya. 2011. A Geometric Approach to Dynamically Feasible, Real-Time Formation Control. *Journal of Dynamic Systems, Measurement, and Control* **133**:2, 021010. [CrossRef]

73. Cesar A. Ocampo, Dennis V. ByrnesMission Design and Trajectory Optimization . [CrossRef]

74. Hendrikus.G. VisserAirplane Performance Optimization . [CrossRef]

75. Giorgio MingottiPini Gurfil. 2010. Mixed Low-Thrust Invariant-Manifold Transfers to Distant Prograde Orbits Around Mars. *Journal of Guidance, Control, and Dynamics* **33**:6, 1753-1764. [Citation] [PDF] [PDF Plus]

76. Wei Kang. 2010. Rate of convergence for the Legendre pseudospectral optimal control of feedback linearizable systems. *Journal of Control Theory and Applications* **8**:4, 391-405. [CrossRef]

77. P. Pergola. 2010. Low-thrust transfer to Backflip orbits. *Advances in Space Research* **46**:10, 1280-1291. [CrossRef]

78. S. Leyendecker, S. Ober-Blöbaum, J. E. Marsden, M. Ortiz. 2010. Discrete mechanics and optimal control for constrained systems. *Optimal Control Applications and Methods* **31**:6, 505-528. [CrossRef]

79. Knut Graichen, Andreas Kugi, Nicolas Petit, Francois Chaplais. 2010. Handling constraints in optimal control with saturation functions and system extension. *Systems & Control Letters* **59**:11, 671-679. [CrossRef]

80. Enrico Bertolazzi, Francesco Biral, Mauro Da Lio, Andrea Saroldi, Fabio Tango. 2010. Supporting Drivers in Keeping Safe Speed and Safe Distance: The SASPENCE Subproject Within the European Framework Programme 6 Integrating Project PReVENT. *IEEE Transactions on Intelligent Transportation Systems* **11**:3, 525-538. [CrossRef]

81. Florian Holzapfel, ; Jakob Lenz, ; Florian Fisch, ; Gottfried Sachs, On the Solution of Bilevel Optimal Control Problems to Increase the Fairness in Air Races . [Citation] [PDF] [PDF Plus]

82. Ryan Russell, ; Gregory Lantoine, A Unified Framework for Robust Optimization of Interplanetary Trajectories . [Citation] [PDF] [PDF Plus]

83. Antonios Tsourdos, ; Alastair Cooke, ; Rick Drury, Negative-g Trajectory Generation Using Quaternion-Based Inverse Dynamics . [Citation] [PDF] [PDF Plus]

84. Pini Gurfil, ; Giorgio Mingotti, Mixed Low-Thrust and Invariant-Manifold Transfers to Unstable Distant Prograde Orbits Around Mars . [Citation] [PDF] [PDF Plus]

85. Kathleen Howell, ; Martin Ozimek, ; Jeffrey Stuart, Optimal, Low-Thrust, Path-Constrained Transfers Between Libration Point Orbits Using Invariant Manifolds . [Citation] [PDF] [PDF Plus]

86. Ying Nan, GuoQiang Huang, YuPing Lu, Ping Gong. 2010. Global 4-D trajectory optimization for spacecraft. *Science China Technological Sciences* **53**:8, 2097-2101. [CrossRef]

87. Yuri Ulybyshev. 2010. Discrete Pseudocontrol Sets for Optimal Control Problems. *Journal of Guidance, Control, and Dynamics* **33**:4, 1133-1142. [Citation] [PDF] [PDF Plus]

88. Bing-nan KangDavid B SpencerShuo TangDan JordanCarol CesnikAndrea SerraniDarryll Pines. 2010. Optimal Periodic Cruise Trajectories via a Two-Level Optimization Method. *Journal of Spacecraft and Rockets* **47**:4, 597-613. [Citation] [PDF] [PDF Plus]

89. Mohammad Maleki, Abdollah Hadi-Vencheh. 2010. Combination of non-classical pseudospectral and direct methods for the solution of brachistochrone problem. *International Journal of Computer Mathematics* **87**:8, 1847-1856. [CrossRef]

90. Carl Glen Henshaw, Robert M. Sanner. 2010. Variational Technique for Spacecraft Trajectory Planning. *Journal of Aerospace Engineering* **23**:3, 147-156. [CrossRef]

91. Stuart A StantonBelinda G Marchand. 2010. Enhanced Collocation Method for Dynamical Systems Subject to Finite Set Control. *Journal of Guidance, Control, and Dynamics* **33**:3, 957-968. [Citation] [PDF] [PDF Plus]
92. Joris T. Olympio. 2010. Optimal Control of Gravity-Tractor Spacecraft for Asteroid Deflection. *Journal of Guidance, Control, and Dynamics* **33**:3, 823-833. [Citation] [PDF] [PDF Plus]
93. Nesrin Sarigul-Klijn, ; Albert Jordan, Statistical and Probabilistic Path Planning for Aircarft with Performance Uncertainities . [Citation] [PDF] [PDF Plus]
94. Marko Ackermann, Antonie J. van den Bogert. 2010. Optimality principles for model-based prediction of human gait. *Journal of Biomechanics* **43**:6, 1055-1060. [CrossRef]
95. G. Mavelli, P. Palumbo. 2010. The Carleman Approximation Approach to Solve a Stochastic Nonlinear Control Problem. *IEEE Transactions on Automatic Control* **55**:4, 976-982. [CrossRef]
96. M. T OzimekK. C HowellDavid C FoltaRobert G MeltonCraig A KlueverThomas Starchville. 2010. Low-Thrust Transfers in the Earth-Moon System, Including Applications to Libration Point Orbits. *Journal of Guidance, Control, and Dynamics* **33**:2, 533-549. [Citation] [PDF] [PDF Plus]
97. Massimiliano MatteiLuciano Blasi. 2010. Smooth Flight Trajectory Planning in the Presence of No-Fly Zones and Obstacles. *Journal of Guidance, Control, and Dynamics* **33**:2, 454-462. [Citation] [PDF] [PDF Plus]
98. Daniel P Scharf, Jason A Keim, Fred Y Hadaegh. 2010. Flight-Like Ground Demonstrations of Precision Maneuvers for Spacecraft Formations—Part I. *IEEE Systems Journal* **4**:1, 84-95. [CrossRef]
99. S.A. Fazelzadeh, G.A. Varzandian. 2010. Minimum-time Earth–Moon and Moon–Earth orbital maneuvers using time-domain finite element method. *Acta Astronautica* **66**:3-4, 528-538. [CrossRef]
100. Nesrin Sarigul-KlijnR. RapettiA. JordanI. LopezM. Sarigul-KlijnP. NespecaRonald HessMike BraggGary VanderplaatsJohn Burken. 2010. Intelligent Flight-Trajectory Generation to Maximize Safe-Outcome Probability After a Distress Event. *Journal of Aircraft* **47**:1, 255-267. [Citation] [PDF] [PDF Plus]
101. Carlo L. Bottasso, Giorgio Maisano, Francesco Scorcelletti. 2010. Trajectory Optimization Procedures for Rotorcraft Vehicles, Their Software Implementation, and Applicability to Models of Increasing Complexity. *Journal of the American Helicopter Society* **55**:3, 032010. [CrossRef]
102. C. Goerzen, Z. Kong, B. Mettler. 2010. A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. *Journal of Intelligent and Robotic Systems* **57**:1-4, 65-100. [CrossRef]
103. John Gregory, Ana García-Bouso, Alberto Olivares, Ernesto Staffetti. 2009. Energy-optimal control of unconstrained planar RR robot manipulators. *TOP* **17**:2, 385-406. [CrossRef]
104. Giovanni Mengali, Alessandro A. Quarta. 2009. Solar sail trajectories with piecewise-constant steering laws. *Aerospace Science and Technology* **13**:8, 431-441. [CrossRef]
105. M. T. OzimekD. J. GrebowK. C. HowellRobert MeltonColin McInnesDavid Folta. 2009. Design of Solar Sail Trajectories with Applications to Lunar South Pole Coverage. *Journal of Guidance, Control, and Dynamics* **32**:6, 1884-1897. [Citation] [PDF] [PDF Plus]
106. Rodney L. AndersonMartin W. LoSeungwon LeeTheodore H. SweetserJeffrey S. ParkerBenjamin VillacRandy Paffenroth. 2009. Role of Invariant Manifolds in Low-Thrust Trajectory Design. *Journal of Guidance, Control, and Dynamics* **32**:6, 1921-1930. [Citation] [PDF] [PDF Plus]
107. Yang Gao. 2009. Linear Feedback Guidance for Low-Thrust Many-Revolution Earth-Orbit Transfers. *Journal of Spacecraft and Rockets* **46**:6, 1320-1325. [Citation] [PDF] [PDF Plus]
108. G. Mingotti, F. Topputo, F. Bernelli-Zazzera. 2009. Low-energy, low-thrust transfers to the Moon. *Celestial Mechanics and Dynamical Astronomy* **105**:1-3, 61-74. [CrossRef]
109. K. Graichen, N. Petit. 2009. Incorporating a class of constraints into the dynamics of optimal control problems. *Optimal Control Applications and Methods* **30**:6, 537-561. [CrossRef]
110. Prashant Patel, Daniel J. Scheeres. 2009. A second-order optimization algorithm using quadric control updates for multistage optimal control problems. *Optimal Control Applications and Methods* **30**:6, 525-536. [CrossRef]
111. Camilla Colombo, Massimiliano Vasile, Gianmarco Radice. 2009. Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming. *Celestial Mechanics and Dynamical Astronomy* **105**:1-3, 75-112. [CrossRef]
112. Michael Dellnitz, Sina Ober-Blöbaum, Marcus Post, Oliver Schütze, Bianca Thiere. 2009. A multi-objective approach to the design of low thrust space trajectories using optimal control. *Celestial Mechanics and Dynamical Astronomy* **105**:1-3, 33-59. [CrossRef]

113. P. Pergola, K. Geurts, C. Casaregola, M. Andrenucci. 2009. Earth–Mars halo to halo low thrust manifold transfers. *Celestial Mechanics and Dynamical Astronomy* **105**:1-3, 19-32. [CrossRef]

114. Qian Wang, Jasbir S. Arora. 2009. Several simultaneous formulations for transient dynamic response optimization: An evaluation. *International Journal for Numerical Methods in Engineering* **80**:5, 631-650. [CrossRef]

115. K.A.C. Baumgartner, S. Ferrari, A.V. Rao. 2009. Optimal Control of an Underwater Sensor Network for Cooperative Target Tracking. *IEEE Journal of Oceanic Engineering* **34**:4, 678-697. [CrossRef]

116. Joseph Z. Ben-AsherDror Cohen. 2009. Allocation of Radar Tracking Resources by Direct Optimization. *Journal of Aerospace Computing, Information, and Communication* **6**:9, 523-539. [Citation] [PDF] [PDF Plus]

117. Nobuhiro YokoyamaYoshimasa Ochi. 2009. Path Planning Algorithms for Skid-to-Turn Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics* **32**:5, 1531-1543. [Citation] [PDF] [PDF Plus]

118. Mauro MassariFranco Bernelli-Zazzera. 2009. Optimization of Low-Thrust Reconfiguration Maneuvers for Spacecraft Flying in Formation. *Journal of Guidance, Control, and Dynamics* **32**:5, 1629-1638. [Citation] [PDF] [PDF Plus]

119. Kelli A. C. Baumgartner, Silvia Ferrari, Thomas A. Wettergren. 2009. Robust Deployment of Dynamic Sensor Networks for Cooperative Track Detection. *IEEE Sensors Journal* **9**:9, 1029-1048. [CrossRef]

120. Joseph Mueller, ; Yiyuan Zhao, ; William Garrard, Sensitivity and Solar Power Analysis of Optimal Trajectories for Autonomous Airships . [Citation] [PDF] [PDF Plus]

121. Yuri Ulybyshev, Discrete Pseudo-Control Sets for Optimal Control Problem . [Citation] [PDF] [PDF Plus]

122. Bingnan Kang, ; David Spencer, ; Shuo Tang, ; Dan Jordan, Study of Optimal Periodic Cruise Trajectories via Tradespace Visualization . [Citation] [PDF] [PDF Plus]

123. Di Wu, ; Yiyuan Zhao, Performances and Sensitivities of Optimal Trajectory Generation for Air Traffic Control Automation . [Citation] [PDF] [PDF Plus]

124. Yunjun Xu, ; Gareth Basset, Pre and Post Optimality Checking of the Virtual Motion Camouflage Based Nonlinear Constrained Subspace Optimal Control . [Citation] [PDF] [PDF Plus]

125. Rick Drury, ; James Whidborne, A Quaternion-Based Inverse Dynamics Model for Real-Time UAV Trajectory Generation . [Citation] [PDF] [PDF Plus]

126. Gao Yang. 2009. Direct Optimization of Low-thrust Many-revolution Earth-orbit Transfers. *Chinese Journal of Aeronautics* **22**:4, 426-433. [CrossRef]

127. Yuri Ulybyshev. 2009. Spacecraft Trajectory Optimization Based on Discrete Sets of Pseudoimpulses. *Journal of Guidance, Control, and Dynamics* **32**:4, 1209-1217. [Citation] [PDF] [PDF Plus]

128. Kamesh SubbaraoBrandon M Shippey. 2009. Hybrid Genetic Algorithm Collocation Method for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics* **32**:4, 1396-1403. [Citation] [PDF] [PDF Plus]

129. Rick G DruryJames F Whidborne. 2009. Quaternion-Based Inverse Dynamics Model for Expressing Aerobatic Aircraft Trajectories. *Journal of Guidance, Control, and Dynamics* **32**:4, 1388-1391. [Citation] [PDF] [PDF Plus]

130. Sung Hyun KimRaktim BhattacharyaElla AtkinsTamer InancPhil Chandler. 2009. Motion Planning in Obstacle Rich Environments. *Journal of Aerospace Computing, Information, and Communication* **6**:7, 433-450. [Citation] [PDF] [PDF Plus]

131. Baolin WuDanwei WangEng Kee PohGuangyan XuJONATHAN P. HOWHui YanYang ChengPeter M. BainumRussell Carpenter. 2009. Nonlinear Optimization of Low-Thrust Trajectory for Satellite Formation: Legendre Pseudospectral Approach. *Journal of Guidance, Control, and Dynamics* **32**:4, 1371-1381. [Citation] [PDF] [PDF Plus]

132. Joseph B. MuellerYiyuan J. ZhaoWilliam L. GarrardAnthony ColozzaJohn SullivanDavid SchmidtPing Lu. 2009. Optimal Ascent Trajectories for Stratospheric Airships Using Wind Energy. *Journal of Guidance, Control, and Dynamics* **32**:4, 1232-1245. [Citation] [PDF] [PDF Plus]

133. Baljeet SinghRaktim BhattacharyaSrinivas R VadaliJohn T BettsI. Michael RossPaul WilliamsHans SeywaldAnil V Rao. 2009. Verification of Optimality and Costate Estimation Using Hilbert Space Projection. *Journal of Guidance, Control, and Dynamics* **32**:4, 1345-1355. [Citation] [PDF] [PDF Plus]

134. Yiyuan J Zhao. 2009. Extracting Energy from Downdraft to Enhance Endurance of Uninhabited Aerial Vehicles. *Journal of Guidance, Control, and Dynamics* **32**:4, 1124-1133. [Citation] [PDF] [PDF Plus]

135. Paul WilliamsJohn BettsBruce Conway. 2009. Hermite-Legendre-Gauss-Lobatto Direct Transcription in Trajectory Optimization. *Journal of Guidance, Control, and Dynamics* **32**:4, 1392-1395. [Citation] [PDF] [PDF Plus]

136. S. Gros, B. Chachuat, D. Bonvin. 2009. Neighbouring-extremal control for singular dynamic optimisation problems. Part II: multiple-input systems. *International Journal of Control* **82**:7, 1193-1211. [CrossRef]

137. Nathan Robert HarlHenry John PernickaMark BeckmanRivers LambDavid FoltaCraig KlueverJohn Junkins. 2009. Low-Thrust Control of a Lunar Mapping Orbit. *Journal of Guidance, Control, and Dynamics* **32**:3, 939-948. [Citation] [PDF] [PDF Plus]

138. Hoam Chung, E. Polak, S. Sastry. 2009. An External Active-Set Strategy for Solving Optimal Control Problems. *IEEE Transactions on Automatic Control* **54**:5, 1129-1133. [CrossRef]

139. ChangQing Chen, YongChun Xie. 2009. Optimal impulsive ellipse-to-circle coplanar rendezvous. *Science in China Series E: Technological Sciences* **52**:5, 1435-1445. [CrossRef]

140. Yiming Zhao, ; Panagiotis Tsiotras, Mesh Refinement Using Density Function for Solving Optimal Control Problems . [Citation] [PDF] [PDF Plus]

141. Wen Zhang, He-ping Ma. 2009. Chebyshev-Legendre method for discretizing optimal control problems. *Journal of Shanghai University (English Edition)* **13**:2, 113-118. [CrossRef]

142. Giovanni MengaliAlessandro A. QuartaLes JohnsonColin R. McInnesBernd DachwaldVictoria L. CoverstoneManfred Leipold. 2009. Solar Sail Near-Optimal Circular Transfers with Plane Change. *Journal of Guidance, Control, and Dynamics* **32**:2, 456-463. [Citation] [PDF] [PDF Plus]

143. Timothy R JorrisRichard G. CobbAnil V RaoDavid DomanMichael W. OppenheimerChristopher S. SchulzFariba Fahroo. 2009. Three-Dimensional Trajectory Optimization Satisfying Waypoint and No-Fly Zone Constraints. *Journal of Guidance, Control, and Dynamics* **32**:2, 551-572. [Citation] [PDF] [PDF Plus]

144. Hao Wen, DongPing Jin, HaiYan Hu. 2009. Costate estimation for dynamic systems of the second order. *Science in China Series E: Technological Sciences* **52**:3, 752-760. [CrossRef]

145. Chang-Joo KimSoo Hyung ParkSang Kyung SungSung-Nam Jung. 2009. Nonlinear Optimal Control Analysis Using State-Dependent Matrix Exponential and Its Integrals. *Journal of Guidance, Control, and Dynamics* **32**:1, 309-313. [Citation] [PDF] [PDF Plus]

146. Brian C. Fabien. 2009. dsoa: The implementation of a dynamic system optimization algorithm. *Optimal Control Applications and Methods* n/a-n/a. [CrossRef]

147. Inseok Hwang, Jinhua Li, Dzung Du. 2009. Differential Transformation and Its Application to Nonlinear Optimal Control. *Journal of Dynamic Systems, Measurement, and Control* **131**:5, 051010. [CrossRef]

148. Yunjun Xu. 2009. Trajectory Analysis for Vertical Takeoff and Vertical Landing Reusable Launch Vehicle's Upper Stage. *Journal of Aerospace Engineering* **22**:1, 58-66. [CrossRef]

149. Sang-Jong Lee, Hyo-Choong Bang, Jae-Won Chang, Kie-Jeong Seong. 2009. Trajectory Optimization for Nonlinear Tracking Control in Stratospheric Airship Platform. *Journal of the Korean Society for Aeronautical & Space Sciences* **37**:1, 42-54. [CrossRef]

150. Georges S. Aoude, Jonathan P. How, Ian M. Garcia. 2008. Two-stage path planning approach for solving multiple spacecraft reconfiguration maneuvers. *The Journal of the Astronautical Sciences* **56**:4, 515-544. [CrossRef]

151. Brian C. Fabien. 2008. Direct optimization of dynamic systems described by differential-algebraic equations. *Optimal Control Applications and Methods* **29**:6, 445-466. [CrossRef]

152. R. Lopez, E. Balsa-Canto, E. Oñate. 2008. Neural networks for variational problems in engineering. *International Journal for Numerical Methods in Engineering* **75**:11, 1341-1360. [CrossRef]

153. Chang-Joo KimSang Kyung SungSoo Hyung ParkSung-Nam JungKwanjung Yee. 2008. Selection of Rotorcraft Models for Application to Optimal Control Problems. *Journal of Guidance, Control, and Dynamics* **31**:5, 1386-1399. [Citation] [PDF] [PDF Plus]

154. Sachin JainPanagiotis Tsiotras. 2008. Trajectory Optimization Using Multiresolution Techniques. *Journal of Guidance, Control, and Dynamics* **31**:5, 1424-1436. [Citation] [PDF] [PDF Plus]

155. Francesco ToppputoAshraf OwisFranco Bernelli-ZazzeraDaniele MortariColin McInnes. 2008. Analytical Solution of Optimal Feedback Control for Radially Accelerated Orbits. *Journal of Guidance, Control, and Dynamics* **31**:5, 1352-1359. [Citation] [PDF] [PDF Plus]

156. Knut GraichenNicolas PetitMehran MesbahiPanagiotis TsiotrasDavid BayardRichard EpenoyMoritz Diehl. 2008. Constructive Methods for Initialization and Handling Mixed State-Input Constraints in Optimal Control. *Journal of Guidance, Control, and Dynamics* **31**:5, 1334-1343. [Citation] [PDF] [PDF Plus]

157. Anders Eriksson. 2008. Optimization in target movement simulations. *Computer Methods in Applied Mechanics and Engineering* **197**:49-50, 4207-4215. [CrossRef]

158. Behcet Acikmese, ; Daniel Scharf, ; Lars Blackmore, ; Aron Wolf, Enhancements on the Convex Programming Based Powered Descent Guidance Algorithm for Mars Landing . [Citation] [PDF] [PDF Plus]

159. Puneet Singla, ; Tarunraj Singh, A Novel Coordinate Transformation for Obstacle Avoidance and Optimal Trajectory Planning . [Citation] [PDF] [PDF Plus]

160. Yuri Ulybyshev, Spacecraft Trajectory Optimization Based on Discrete Sets of Pseudo-Impulses . [Citation] [PDF] [PDF Plus]

161. Gregory Lantoine, ; Ryan Russell, A Hybrid Differential Dynamic Programming Algorithm for Robust Low-Thrust Optimization . [Citation] [PDF] [PDF Plus]

162. Matthew Vavrina, ; Kathleen Howell, Global Low-Thrust Trajectory Optimization Through Hybridization of a Genetic Algorithm and a Direct Method . [Citation] [PDF] [PDF Plus]

163. Rosemary Huang, ; Inseok Hwang, ; Martin Corless, Application of Differential Transformation Based Algorithm to Find Optimal Interplanetary Low-Thrust Transfers . [Citation] [PDF] [PDF Plus]

164. Nobuhiro Yokoyama, ; Yoshimasa Ochi, Optimal Path Planning for Skid-to-Turn Unmanned Aerial Vehicle . [Citation] [PDF] [PDF Plus]

165. Martin Ozimek, ; Kahtleen Howell, ; Daniel Grebow, Solar Sails and Lunar South Pole Coverage . [Citation] [PDF] [PDF Plus]

166. R. Lopez, E. Oñate. 2008. An extended class of multilayer perceptron. *Neurocomputing* **71**:13-15, 2538-2543. [CrossRef]

167. C.Z. Wu, K.L. Teo, Yi Zhao. 2008. Numerical method for a class of optimal control problems subject to nonsmooth functional constraints. *Journal of Computational and Applied Mathematics* **217**:2, 311-325. [CrossRef]

168. Asif FarooqDavid John Limebeer. 2008. Optimal Trajectory Regulation for Radar Imaging Guidance. *Journal of Guidance, Control, and Dynamics* **31**:4, 1076-1092. [Citation] [PDF] [PDF Plus]

169. Janne KarelahtiKai VirtanenJohn ÖströmJohn T. BettsRonald A. HessKlaus WellUlf RingertzFumiaki Imado. 2008. Automated Generation of Realistic Near-Optimal Aircraft Trajectories. *Journal of Guidance, Control, and Dynamics* **31**:3, 674-688. [Citation] [PDF] [PDF Plus]

170. Geoffrey T HuntingtonAnil V RaoScott R PloenWayne P HallmanVictoria Coverstone. 2008. Optimal Reconfiguration of Spacecraft Formations Using the Gauss Pseudospectral Method. *Journal of Guidance, Control, and Dynamics* **31**:3, 689-698. [Citation] [PDF] [PDF Plus]

171. Qi GongFariba FahrooI. Michael Ross. 2008. Spectral Algorithm for Pseudospectral Methods in Optimal Control. *Journal of Guidance, Control, and Dynamics* **31**:3, 460-471. [Citation] [PDF] [PDF Plus]

172. François Chaplais, Nicolas Petit. 2008. Inversion in indirect optimal control of multivariable systems. *ESAIM: Control, Optimisation and Calculus of Variations* **14**:2, 294-317. [CrossRef]

173. Abolghasem Naghash, Reza Esmaelzadeh, Mehdi Mortazavi, Reza Jamilnia. 2008. Near optimal guidance law for descent to a point using inverse problem approach. *Aerospace Science and Technology* **12**:3, 241-247. [CrossRef]

174. Geoffrey T HuntingtonAnil V RaoLorenz BieglerJuan Arrieta-CamachoWayne P Hallman. 2008. Comparison of Global and Local Collocation Methods for Optimal Control. *Journal of Guidance, Control, and Dynamics* **31**:2, 432-436. [Citation] [PDF] [PDF Plus]

175. R. Esmaelzadeh, A. Naghash, M. Mortazavi. 2008. Near optimal re-entry guidance law using inverse problem approach. *Inverse Problems in Science and Engineering* **16**:2, 187-198. [CrossRef]

176. H. Wen, D. P. Jin, H. Y. Hu. 2008. Optimal feedback control of the deployment of a tethered subsatellite subject to perturbations. *Nonlinear Dynamics* **51**:4, 501-514. [CrossRef]

177. N. Yokoyama, S. Suzuki, T. Tsuchiya. 2008. Convergence Acceleration of Direct Trajectory Optimization Using Novel Hessian Calculation Methods. *Journal of Optimization Theory and Applications* **136**:3, 297-319. [CrossRef]

178. Manindra Kaphle, Anders Eriksson. 2008. Optimality in forward dynamics simulations. *Journal of Biomechanics* **41**:6, 1213-1221. [CrossRef]

179. Zhanhua Ma, Ou Ma, Banavara N. Shashikanth. 2007. Optimal approach to and alignment with a rotating rigid body for capture. *The Journal of the Astronautical Sciences* **55**:4, 407-419. [CrossRef]

180. 2007. Effect of the spanwise grid spacing and treatment of convection term in DES. *International Journal of Aeronautical and Space Sciences* **8**:2, 1-10. [CrossRef]

181. Carlos Corral van Damme, Tomás Prieto-Llanos, Raúl Cadenas, Jesús Gil-Fernández, Mariella Graziano. 2007. 1st ACT global trajectory optimization competition: Results found at GMV. *Acta Astronautica* **61**:9, 786-793. [CrossRef]

182. 2007. Minimum-Time Attitude Reorientations of Three-Axis Stabilized Spacecraft Using Only Magnetic Torquers. *International Journal of Aeronautical and Space Sciences* **8**:2, 17-27. [CrossRef]

183. W. Kang, Q. Gong, I. M. Ross, F. Fahroo. 2007. On the convergence of nonlinear optimal control using pseudospectral methods for feedback linearizable systems. *International Journal of Robust and Nonlinear Control* **17**:14, 1251-1277. [CrossRef]

184. Julien Laurent-VarinNicolas BérendFrédéric BonnansMounir HaddouChristophe TalbotJohn T. BettsKlaus WellHans Josef Pesch. 2007. Interior-Point Approach to Trajectory Optimization. *Journal of Guidance, Control, and Dynamics* **30**:5, 1228-1238. [Citation] [PDF] [PDF Plus]

185. Urbano TancrediMichele GrassiEnrico LorenziniZuojun ShenClemens Tillier. 2007. Approximate Trajectories for Thermal Protection System Flight Tests Mission Design. *Journal of Spacecraft and Rockets* **44**:5, 1003-1011. [Citation] [PDF] [PDF Plus]

186. Sangjong LeeHyochoong Bang. 2007. Three-Dimensional Ascent Trajectory Optimization for Stratospheric Airship Platforms in the Jet Stream. *Journal of Guidance, Control, and Dynamics* **30**:5, 1341-1351. [Citation] [PDF] [PDF Plus]

187. Janne KarelahtiKai VirtanenTuomas RaivioMichael BreitnerLeena SinghUlf RingertzFumiaki Imado. 2007. Near-Optimal Missile Avoidance Trajectories via Receding Horizon Control. *Journal of Guidance, Control, and Dynamics* **30**:5, 1287-1298. [Citation] [PDF] [PDF Plus]

188. Jacob DemeyerPini GurfilCesar OcampoDavid Folta. 2007. Transfer to Distant Retrograde Orbits Using Manifold Theory. *Journal of Guidance, Control, and Dynamics* **30**:5, 1261-1267. [Citation] [PDF] [PDF Plus]

189. Behcet AcikmeseScott R PloenKenneth MeaseMehran MesbahiCornel SultanAnastassios E PetropoulosPing Lu. 2007. Convex Programming Approach to Powered Descent Guidance for Mars Landing. *Journal of Guidance, Control, and Dynamics* **30**:5, 1353-1366. [Citation] [PDF] [PDF Plus]

190. 2007. Low Thrust, Fuel Optimal Earth Escape Trajectories Design. *Journal of the Korean Society for Aeronautical & Space Sciences* **35**:7, 647-654. [CrossRef]

191. I. Michael RossQi GongPooya SekhavatVictoria CoverstoneRobert MeltonKyle AlfriendDavid Vallado. 2007. Low-Thrust, High-Accuracy Trajectory Optimization. *Journal of Guidance, Control, and Dynamics* **30**:4, 921-933. [Citation] [PDF] [PDF Plus]

192. Geoffrey T. Huntington, David Benson, Anil V. Rao. 2007. Optimal configuration of tetrahedral spacecraft formations. *The Journal of the Astronautical Sciences* **55**:2, 141-169. [CrossRef]

193. Rong Zhu, Dong Sun, Zhaoying Zhou. 2007. Integrated design of trajectory planning and control for micro air vehicles. *Mechatronics* **17**:4-5, 245-253. [CrossRef]

194. Clment Petres, Yan Pailhas, Pedro Patron, Yvan Petillot, Jonathan Evans, David Lane. 2007. Path Planning for Autonomous Underwater Vehicles. *IEEE Transactions on Robotics* **23**:2, 331-341. [CrossRef]

195. Enrico Bertolazzi, Francesco Biral, Mauro Da Lio. 2007. real-time motion planning for multibody systems. *Multibody System Dynamics* **17**:2-3, 119-139. [CrossRef]

196. Yuri Ulybyshev. 2007. Continuous Thrust Orbit Transfer Optimization Using Large-Scale Linear Programming. *Journal of Guidance, Control, and Dynamics* **30**:2, 427-436. [Citation] [PDF] [PDF Plus]

197. Yunjun XuNorman Fitz-CoySimone D'AmicoJohn HansonPaul MasonMason BeckPing Lu. 2007. Enhancement in Optimal Multiple-Burn Trajectory Computation by Switching Function Analysis. *Journal of Spacecraft and Rockets* **44**:1, 264-272. [Citation] [PDF] [PDF Plus]

198. Ya-zhong LuoHai-yang LIGuo-jin Tang. 2007. Hybrid Approach to Optimize a Rendezvous Phasing Strategy. *Journal of Guidance, Control, and Dynamics* **30**:1, 185-191. [Citation] [PDF] [PDF Plus]

199. Yuan Ren, Pingyuan Cui, Enjie Luan. 2007. A novel algorithm to optimize complicated low-thrust trajectory. *Aircraft Engineering and Aerospace Technology* **79**:3, 283-288. [CrossRef]

200. I. Michael Ross6 Space trajectory optimization and L1-optimal control problems **1**, 155-VIII. [CrossRef]

201. D. H. A. Maithripala, Suhada Jayasuriya, Mark J. Mears. 2007. Phantom Track Generation Through Cooperative Control of Multiple ECAVs Based on Feasibility Analysis. *Journal of Dynamic Systems, Measurement, and Control* **129**:5, 708. [CrossRef]

202. Vincent M. Guibout *, Daniel J. Scheeres †3 Solving Two-Point Boundary Value Problems Using Generating Functions: Theory and Applications to Astrodynamics **1**, 53-105. [CrossRef]

203. Djaffar Boussaa. 2006. Optimizing the composition profile of a functionally graded interlayer using a direct transcription method. *Computational Mechanics* **39**:1, 59-71. [CrossRef]

204. David A BensonGeoffrey T HuntingtonAnil V RaoTom P ThorvaldsenJohn T BettsStephen W ParisKenneth D MeaseWayne P Hallman. 2006. Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method. *Journal of Guidance, Control, and Dynamics* **29**:6, 1435-1440. [Citation] [PDF] [PDF Plus]

205. Fabrizio GiuliettiAlessandro A. QuartaPaolo TortoraCraig A. KlueverBong WieYakov OshmanChristopher D. HallMark L. Psiaki. 2006. Optimal Control Laws for Momentum-Wheel Desaturation Using Magnetorquers. *Journal of Guidance, Control, and Dynamics* **29**:6, 1464-1468. [Citation] [PDF] [PDF Plus]

206. S. Vougioukas, S. Blackmore, J. Nielsen, S. Fountas. 2006. A two-stage optimal motion planner for autonomous agricultural vehicles. *Precision Agriculture* **7**:5, 361-377. [CrossRef]

207. Ya-Zhong Luo, Guo-Jin Tang. 2006. Rendezvous phasing special-point maneuvers mixed discrete-continuous optimization using simulated annealing. *Aerospace Science and Technology* **10**:7, 652-658. [CrossRef]

208. Daewoo Lee. 2006. Optimization analysis of trajectory for re-entry vehicle using global orthogonal polynomial. *Journal of Mechanical Science and Technology* **20**:10, 1557-1566. [CrossRef]

209. R. Armellin, M. Lavagna, A. Ercoli-Finzi. 2006. Aero-gravity assist maneuvers: controlled dynamics modeling and optimization. *Celestial Mechanics and Dynamical Astronomy* **95**:1-4, 391-405. [CrossRef]

210. Robert WindhorstEric GallowayEric LauDavid SaundersPeter GageJeffrey BowlesMark ArdemaP. K. MenonGano ChatterjiDavid Kinney. 2006. Aerospace Vehicle Trajectory Design and Optimization Within a Multi-Disciplinary Environment. *Journal of Aerospace Computing, Information, and Communication* **3**:9, 471-485. [Citation] [PDF] [PDF Plus]

211. P. De PascaleM. VasileAnastassios PetropoulosCesar OcampoGianmarco RadiceStephen KembleDilmurat Azimov. 2006. Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories. *Journal of Spacecraft and Rockets* **43**:5, 1065-1076. [Citation] [PDF] [PDF Plus]

212. Ya-Zhong Luo, Guo-Jin Tang, Hai-yang Li. 2006. Optimization of multiple-impulse minimum-time rendezvous with impulse constraints using a hybrid genetic algorithm. *Aerospace Science and Technology* **10**:6, 534-540. [CrossRef]

213. Ella M AtkinsIgor Alonso PortilloMatthew J StrubeNhan NguyenJames NeidhoeferMehmet Ertem. 2006. Emergency Flight Planning Applied to Total Loss of Thrust. *Journal of Aircraft* **43**:4, 1205-1216. [Citation] [PDF] [PDF Plus]

214. John T. Betts. 2006. Trajectory optimization in the presence of uncertainty. *The Journal of the Astronautical Sciences* **54**:2, 227-243. [CrossRef]

215. Mateen-ud-Din Qazi, He Linshu. 2006. Nearly-orthogonal sampling and neural network metamodel driven conceptual design of multistage space launch vehicle. *Computer-Aided Design* **38**:6, 595-607. [CrossRef]

216. 2006. Dynamic Equations of Motion and Trajectory Optimization for the Mid-Altitude Unmanned Airship Platform. *Journal of the Korean Society for Aeronautical & Space Sciences* **34**:5, 46-55. [CrossRef]

217. Roger L BarronCleon M Chick IIIPing LuDavid G WardJohn SchiermanRussell EnnsAnthony J Calise. 2006. Improved Indirect Method for Air-Vehicle Trajectory Optimization. *Journal of Guidance, Control, and Dynamics* **29**:3, 643-652. [Citation] [PDF] [PDF Plus]

218. I. Michael Ross, Fariba Fahroo. 2006. Issues in the real-time computation of optimal control. *Mathematical and Computer Modelling* **43**:9-10, 1172-1188. [CrossRef]

219. Nicolas Berend, Christophe Talbot. 2006. Overview of some optimal control methods adapted to expendable and reusable launch vehicle trajectories. *Aerospace Science and Technology* **10**:3, 222-232. [CrossRef]

220. Nedim M. Alemdar, Sibel Sirakaya, Farhad Hüsseinov. 2006. Optimal time aggregation of infinite horizon control problems. *Journal of Economic Dynamics and Control* **30**:4, 569-593. [CrossRef]
221. Ryan P RussellCesar A OcampoTroy McConaghyChauncey UphoffJohn NiehoffCraig A. KlueverDavid Spencer. 2006. Optimization of a Broad Class of Ephemeris Model Earth-Mars Cyclers. *Journal of Guidance, Control, and Dynamics* **29**:2, 354-367. [Citation] [PDF] [PDF Plus]
222. Chris DeverBernard MettlerEric FeronJovan PopovicMarc McConleyJohn HauserAlberto BemporadStefan ShaalFrancesco BulloFrancesco Borrelli. 2006. Nonlinear Trajectory Generation for Autonomous Vehicles via Parametrized Maneuver Classes. *Journal of Guidance, Control, and Dynamics* **29**:2, 289-302. [Citation] [PDF] [PDF Plus]
223. C. Büskens, M. Knauer. 2006. Higher Order Real-Time Approximations in Optimal Control of Multibody-Systems for Industrial Robots. *Multibody System Dynamics* **15**:1, 85-106. [CrossRef]
224. Shunsuke Imamura, Hirohisa Kojima, Takeshi Tsuchiya, Hirotoshi Kubota. 2006. Optimal Design of Two Stage to Orbit Spaceplane Using Hybrid Genetic Algorithm. *JOURNAL OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES* **54**:630, 279-287. [CrossRef]
225. Paul WilliamsPeter BainumEnrico LorenziniHironori A FujiiHirohisa KojimaJesus Pelaez. 2005. Spacecraft Rendezvous on Small Relative Inclination Orbits Using Tethers. *Journal of Spacecraft and Rockets* **42**:6, 1047-1060. [Citation] [PDF] [PDF Plus]
226. Paul Williams, Chris Blanksby, Pavel Trivailo, Hironori A. Fujii. 2005. In-plane payload capture using tethers. *Acta Astronautica* **57**:10, 772-787. [CrossRef]
227. Qian WangJasbir S. AroraG.J. ParkDonald E. GriersonRamana GrandhiRick Balling. 2005. Alternative Formulations for Transient Dynamic Response Optimization. *AIAA Journal* **43**:10, 2188-2195. [Citation] [PDF] [PDF Plus]
228. Gwanyoung Moon, Youdan Kim. 2005. Flight path optimization passing through waypoints for autonomous flight control systems. *Engineering Optimization* **37**:7, 755-774. [CrossRef]
229. Rong ZhuDong SunZhaoying Zhou. 2005. Cooperation Strategy of Unmanned Air Vehicles for Multitarget Interception. *Journal of Guidance, Control, and Dynamics* **28**:5, 1068-1072. [Citation] [PDF] [PDF Plus]
230. H.G. Visser. 2005. Generic and site-specific criteria in the optimization of noise abatement trajectories. *Transportation Research Part D: Transport and Environment* **10**:5, 405-419. [CrossRef]
231. I. Michael RossChristopher N. D'SouzaPing LuBruce ConwayDavid DomanJohn HansonSteve Paris. 2005. Hybrid Optimal Control Framework for Mission Planning. *Journal of Guidance, Control, and Dynamics* **28**:4, 686-697. [Citation] [PDF] [PDF Plus]
232. Martin NorsellDaniel P RaymerRichard M MurrayChrister LarssonJohn T Betts. 2005. Multistage Trajectory Optimization with Radar Range Constraints. *Journal of Aircraft* **42**:4, 849-857. [Citation] [PDF] [PDF Plus]
233. Nahum ShimkinAharon Bar-GillEran RippelJason L SpeyerEric FeronRobert F StengelJohn N Tsitsiklis. 2005. Fast Graph-Search Algorithms for General-Aviation Flight Trajectory Generation. *Journal of Guidance, Control, and Dynamics* **28**:4, 801-811. [Citation] [PDF] [PDF Plus]
234. T. Chettibi, M. Haddad, A. Labed, S. Hanchi. 2005. Generating optimal dynamic motions for closed-chain robotic systems. *European Journal of Mechanics - A/Solids* **24**:3, 504-518. [CrossRef]
235. Ying "Celia" Qi, Yiyuan J. Zhao. 2005. Energy-Efficient Trajectories of Unmanned Aerial Vehicles Flying through Thermals. *Journal of Aerospace Engineering* **18**:2, 84-92. [CrossRef]
236. Paul WilliamsSteven G TragesserJesus PelaezHironori A FujiiEnrico Lorenzini. 2005. Optimal Orbit Transfer with Electrodynamic Tether. *Journal of Guidance, Control, and Dynamics* **28**:2, 369-372. [Citation] [PDF] [PDF Plus]
237. Nobuhiro YokoyamaShinji Suzuki. 2005. Modified Genetic Algorithm for Constrained Trajectory Optimization. *Journal of Guidance, Control, and Dynamics* **28**:1, 139-144. [Citation] [PDF] [PDF Plus]
238. Hiroaki UmeharaColin R. McInnesPhil PalmerFrank McQuadeShinichi NakasukaMehran Mesbahi. 2005. Penalty-Function Guidance for Multiple-Satellite Cluster Formation. *Journal of Guidance, Control, and Dynamics* **28**:1, 182-185. [Citation] [PDF] [PDF Plus]
239. Carlo L. Bottasso, Alessandro Croce, Domenico Leonello, Luca Riviello. 2005. Optimization of Critical Trajectories for Rotorcraft Vehicles. *Journal of the American Helicopter Society* **50**:2, 165. [CrossRef]
240. James M. LonguskiAnastassios E. Petropoulos. 2004. Shape-Based Algorithm for the Automated Design of Low-Thrust, Gravity Assist Trajectories. *Journal of Spacecraft and Rockets* **41**:5, 787-796. [Citation] [PDF] [PDF Plus]

241. Ozan TekinalpMuge BingolPing Lu ProfessorShinji Suzuki ProfessorDavid G. Hull ProfessorAchille Messac ProfessorHajela Prabhat Professor. 2004. Simulated Annealing for Missile Optimization: Developing Method and Formulation Techniques. *Journal of Guidance, Control, and Dynamics* **27**:4, 616-626. [Citation] [PDF] [PDF Plus]

242. Kai VirtanenTuomas RaivioRaimo P. Hamalainen. 2004. Modeling Pilot's Sequential Maneuvering Decisions by a Multistage Influence Diagram. *Journal of Guidance, Control, and Dynamics* **27**:4, 665-677. [Citation] [PDF] [PDF Plus]

243. Ella M AtkinsMin XueJim NeidhoeferJohn Zuk. 2004. Noise-Sensitive Final Approach Trajectory Optimization for Runway-Independent Aircraft. *Journal of Aerospace Computing, Information, and Communication* **1**:7, 269-287. [Citation] [PDF] [PDF Plus]

244. Paul WilliamsChris BlanksbyPavel TrivailoArun MisraEnrico LorenziniPeter BainumChristopher HallVinod Modi. 2004. Tethered Planetary Capture Maneuvers. *Journal of Spacecraft and Rockets* **41**:4, 603-613. [Citation] [PDF] [PDF Plus]

245. T. Chettibi, H.E. Lehtihet, M. Haddad, S. Hanchi. 2004. Minimum cost trajectory planning for industrial robots. *European Journal of Mechanics - A/Solids* **23**:4, 703-715. [CrossRef]

246. I. Michael RossFariba FahrooVictoria Coverstonesteve parisjohn bettsPing Lurobert melton. 2004. Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems. *Journal of Guidance, Control, and Dynamics* **27**:3, 397-405. [Citation] [PDF] [PDF Plus]

247. J.-H.R. Kim, G.L. Lippi, H. Maurer. 2004. Minimizing the transition time in lasers by optimal control methods. *Physica D: Nonlinear Phenomena* **191**:3-4, 238-260. [CrossRef]

248. Paul WilliamsFariba FahrooI M Ross. 2004. Jacobi Pseudospectral Method for Solving Optimal Control Problems. *Journal of Guidance, Control, and Dynamics* **27**:2, 293-297. [Citation] [PDF] [PDF Plus]

249. I. I. HusseinD. J. ScheeresD. C. Hyland. 2004. Interferometric Observatories in Earth Orbit. *Journal of Guidance, Control, and Dynamics* **27**:2, 297-301. [Citation] [PDF] [PDF Plus]

250. Paul WilliamsJames LonguskiSteven TragesserHironori A FujiiArun Misra. 2004. Optimal Control of Tethered Planetary Capture Missions. *Journal of Spacecraft and Rockets* **41**:2, 315-319. [Citation] [PDF] [PDF Plus]

251. Hideaki SHIOIRI, Seiya UENO. 2004. Collision Avoidance Control Law for Aircraft under Uncertain Information. *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES* **47**:157, 209-215. [CrossRef]

252. Eric B. Carlson, Yiyuan J. Zhao. 2004. Optimal City-Center Takeoff Operation of Tiltrotor Aircraft in One Engine Failure. *Journal of Aerospace Engineering* **17**:1, 26-39. [CrossRef]

253. Ali A. Jhemi, Eric B. Carlson, Yiyuan J. Zhao, Robert T. N. Chen. 2004. Optimization of Rotorcraft Flight Following Engine Failure. *Journal of the American Helicopter Society* **49**:2, 117. [CrossRef]

254. Adam Wuerl; Ellen Braden; Tim Crain. 2003. Genetic Algorithm and Calculus of Variations-Based Trajectory Optimization Technique. *Journal of Spacecraft and Rockets* **40**:6, 882-888. [Citation] [PDF] [PDF Plus]

255. Adam WuerlTim CrainEllen Braden. 2003. Genetic Algorithm and Calculus of Variations-Based Trajectory Optimization Technique. *Journal of Spacecraft and Rockets* **40**:6, 882-888. [Citation] [PDF] [PDF Plus]

256. Gi-Yeong Choe. 2003. Generation of an Optimal Trajectory for Rotorcraft Subject to Multiple Waypoint Constraints. *Journal of the Korean Society for Aeronautical & Space Sciences* **31**:8, 50-57. [CrossRef]

257. Eric B. Carlson; Yiyuan J. Zhao. 2003. Prediction of Tiltrotor Height-Velocity Diagrams Using Optimal Control Theory. *Journal of Aircraft* **40**:5, 896-905. [Citation] [PDF] [PDF Plus]

258. J. F. Bonnans, Th. Guilbaud. 2003. Using logarithmic penalties in the shooting algorithm for optimal control problems. *Optimal Control Applications and Methods* **24**:5, 257-278. [CrossRef]

259. Paul Williams, Chris Blanksby, Pavel Trivailo. 2003. Tethered planetary capture: Controlled maneuvers. *Acta Astronautica* **53**:4-10, 681-708. [CrossRef]

260. Hideaki Shioiri, Seiya Ueno. 2003. Collision Avoidance Control Law of Aircraft under Uncertain Information. *JOURNAL OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES* **51**:595, 427-432. [CrossRef]

261. Nobuhiro Yokoyama, Shinji Suzuki. 2003. A Numerical Method for Optimal Control Problem Using Genetic Algorithm. *JOURNAL OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES* **51**:592, 207-214. [CrossRef]

262. Franco Bernelli-Zazzera; Gerlando Cappello; Paolo Mantegazza. 2002. Inverse Dynamics of Aircraft via Reduced-Order Shooting Methods. *Journal of Guidance, Control, and Dynamics* **25**:6, 1158-1162. [Citation] [PDF] [PDF Plus]

263. Anil V. Rao, Sean Tang, Wayne P. Hallman. 2002. Numerical optimization study of multiple-pass aeroassisted orbital transfer. *Optimal Control Applications and Methods* **23**:4, 215-238. [CrossRef]

264. Hussein Jaddu. 2002. Direct solution of nonlinear optimal control problems using quasilinearization and Chebyshev polynomials. *Journal of the Franklin Institute* **339**:4-5, 479-498. [CrossRef]

265. R. Bertrand, R. Epenoy. 2002. New smoothing techniques for solving bang-bang optimal control problems? numerical results and statistical interpretation. *Optimal Control Applications and Methods* **23**:4, 171-197. [CrossRef]

266. 2002. Book Review. *Journal of Guidance, Control, and Dynamics* **25**:3, 607-608. [Citation] [PDF] [PDF Plus]

267. Matthew L. Kaplan, Jean H. Heegaard. 2002. Second-order optimal control algorithm for complex systems. *International Journal for Numerical Methods in Engineering* **53**:9, 2043-2060. [CrossRef]

268. Eric B. Carlson; Yiyuan J. Zhao. 2002. Optimal Short Takeoff of Tiltrotor Aircraft in One Engine Failure. *Journal of Aircraft* **39**:2, 280-289. [Citation] [PDF] [PDF Plus]

269. Eva Balsa Canto, Julio R. Banga, Antonio A. Alonso, Vassilios S. Vassiliadis. 2002. Restricted second order information for the solution of optimal control problems using control vector parameterization. *Journal of Process Control* **12**:2, 243-255. [CrossRef]

270. Fariba Fahroo; I. Michael Ross. 2002. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *Journal of Guidance, Control, and Dynamics* **25**:1, 160-166. [Citation] [PDF] [PDF Plus]

271. Hirohisa Kojima, Nobuyuki Nakajima, Hironori A. Fujii. 2002. Minimum Bending-Moment Control for Slew Maneuver of Flexible Space Structure. Analysis by Hierarchical Gradient Algorithm and Experiment. *JOURNAL OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES* **50**:585, 387-393. [CrossRef]

272. Michael Drumheller. 2002. Constraint-Based Design of Optimal Transport Elements. *Journal of Computing and Information Science in Engineering* **2**:4, 302. [CrossRef]

273. Anthony J. Calise; D. Subbaram Naidu. 2001. Singular Perturbations and Time Scales in Guidance and Control of Aerospace Systems: A Survey. *Journal of Guidance, Control, and Dynamics* **24**:6, 1057-1078. [Citation] [PDF] [PDF Plus]

274. Tuomas Raivio. 2001. Capture Set Computation of an Optimally Guided Missile. *Journal of Guidance, Control, and Dynamics* **24**:6, 1167-1175. [Citation] [PDF] [PDF Plus]

275. J.-H.R. Kim, H. Maurer, Yu.A. Astrov, M. Bode, H.-G. Purwins. 2001. High-Speed Switch-On of a Semiconductor Gas Discharge Image Converter Using Optimal Control Methods. *Journal of Computational Physics* **170**:1, 395-414. [CrossRef]

276. Radu Serban, Linda R. Petzold. 2001. COOPT — a software package for optimal control of large-scale differential–algebraic equation systems. *Mathematics and Computers in Simulation* **56**:2, 187-203. [CrossRef]

277. C.-Y. E. Wang, W.K. Timoszyk, J.E. Bobrow. 2001. Payload maximization for open chained manipulators: finding weightlifting motions for a Puma 762 robot. *IEEE Transactions on Robotics and Automation* **17**:2, 218-224. [CrossRef]

278. Vincent H. Kuo; Yiyuan J. Zhao. 2001. Required Ranges for Conflict Resolutions in Air Traffic Management. *Journal of Guidance, Control, and Dynamics* **24**:2, 237-245. [Citation] [PDF] [PDF Plus]

279. Fariba Fahroo; I. Michael Ross. 2001. Costate Estimation by a Legendre Pseudospectral Method. *Journal of Guidance, Control, and Dynamics* **24**:2, 270-277. [Citation] [PDF] [PDF Plus]

280. Angelo; Daniele Di Bona; Edmondo Minisci; Luciano Guerra; Salvatore D'. 2000. Optimization Methodology for Ascent Trajectories of Lifting-Body Reusable Launchers. *Journal of Spacecraft and Rockets* **37**:6, 761-767. [Citation] [PDF] [PDF Plus]

281. Oleg A. Yakimenko. 2000. Direct Method for Rapid Prototyping of Near-Optimal Aircraft Trajectories. *Journal of Guidance, Control, and Dynamics* **23**:5, 865-875. [Citation] [PDF] [PDF Plus]

282. 2000. Book Review. *Journal of Guidance, Control, and Dynamics* **23**:3, 575-576. [Citation] [PDF] [PDF Plus]

283. Roberto Celi. 2000. Optimization-Based Inverse Simulation of a Helicopter Slalom Maneuver. *Journal of Guidance, Control, and Dynamics* **23**:2, 289-297. [Citation] [PDF] [PDF Plus]

284. Anil V. Rao, Kenneth D. Mease. 2000. Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems. *Optimal Control Applications and Methods* **21**:1, 1-19. [CrossRef]

285. Alexander Heim, Oskar Von Stryk. 2000. Trajectory optimization of industrial robots with application to computer-aided robotics and robot controllers. *Optimization* **47**:3-4, 407-420. [CrossRef]

286. M. Shahzad, J.G. Slama, F. Mora-CaminoA Neural Adaptive Approach for Relative Guidance of Aircraft 123-130. [CrossRef]

287. Chang-Hee Won. 1999. Fuel- or Time-Optimal Transfers Between Coplanar, Coaxial Ellipses Using Lambert's Theorem. *Journal of Guidance, Control, and Dynamics* **22**:4, 536-542. [Citation] [PDF] [PDF Plus]

288. Anil V. Rao, Kenneth D. Mease. 1999. Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems. *Optimal Control Applications and Methods* **20**:2, 59-77. [CrossRef]

289. dé; G. Launay; J. Fré; ric Bonnans. 1998. Large-Scale Direct Optimal Control Applied to a Re-Entry Problem. *Journal of Guidance, Control, and Dynamics* **21**:6, 996-1000. [Citation] [PDF] [PDF Plus]