Annals of the New York Academy of Sciences
Vol. 1065, pp. 210-231, January 2006

A Roadmap for Optimal Control: The Right Way to Commute

I. Michael Ross *
Naval Postgraduate School, Monterey, CA 93943

Optimal control theory is the foundation for many problems in astrodynamics. Typical examples are trajectory design and optimization, relative motion control of distributed space systems and attitude steering. Many such problems in astrodynamics are solved by an alternative route of mathematical analysis and deep physical insight, in part because of the perception that an optimal control framework generates hard problems. While this is indeed true of the Bellman and Pontryagin frameworks, the covector mapping principle provides a neo-classical approach that renders hard problems easy. That is, although the origins of this philosophy can be traced back to Bernoulli and Euler, it is essentially modern as a result of the strong linkage between approximation theory, set-valued analysis and computing technology. Motivated by the broad success of this approach, mission planners are now conceiving and demanding higher performance from space systems. This has resulted in new set of theoretical and computational problems. Recently, under the leadership of NASA-GRC, several workshops were held to address some of these problems. This paper outlines the theoretical issues stemming from practical problems in astrodynamics. Emphasis is placed on how it pertains to advanced mission design problems.

1. Introduction

A vast number of classical and emerging problems in astrodynamics can be most naturally described under the framework of optimal control theory. Lawden[18] pioneered this approach for space trajectory optimization using the older theory of the calculus of variations. The crowing achievement of this approach was his *primer vector theory* which is still used to solved high-thrust trajectory optimization problems. In migrating towards solving low-thrust and other problems in astrodynamics, many practitioners rediscovered the old difficulties of solving optimal control problems. The two classical frameworks are the well-known methods of Bellman and Pontryagin. Refs. [4] and [33] provide recent updates to their methods. In a nutshell, the Bellman approach is beset with many theoretical and practical problems such as the nonsmoothness of the value function and the curse of dimensionality, while the Pontryagin approach generates an "impregnable fortress" (in the words of Poincaré[7]) arising from the symplectic structure of the Hamiltonian system. Although the Bellman and Pontryagin methods are related to one another through a sensitivity relationship, each approach leads to a different set of difficult problems as summarized in Sec. 4. While the details of the ideas and techniques for these approaches are widely different, both approaches perform four common steps towards a proposal for constructing solutions for an abstract optimal control problem, P:

H Hypothesis: Assume a solution exists for Problem P and its perturbations;

$\mathcal{A}$ Approximate: Perturb the solution and generate various approximations;

$\mathcal{L}$ Take limits: Generate limiting conditions – these are the optimality conditions;

$\mathcal{S}$ Solve: Solve for the limiting conditions.

Unlike the hypothesis step, the last three steps are operations; hence, the process can thus be summarized as,

$$\mathcal{SLA}(P)$$

Of course, the details of $\mathcal{S}, \mathcal{L}$ and $\mathcal{A}$ in either the Bellman or the Pontryagin approaches are wildly different. Considering that both approaches generate difficult problems at the end, it is reasonable to argue that the commonality of these approaches may be the root cause of the problems. As a matter of fact, if we commute the last two steps to write,

$$\mathcal{LSA}(P)$$

the resulting problems shift dramatically in rendering hard problems easy. The details of this new philosophy is encapsulated under the *Covector Mapping Principle* (CMP) and is summarized in Sec. 6. Just as the details of $\mathcal{S}, \mathcal{L}$ and $\mathcal{A}$ operations in the Bellman and the Pontryagin frameworks are different, so is the case in the CMP approach. Furthermore, in the spirit of the sensitivity relationship between the Bellman and Pontryagin methods, the CMP framework demands a similar relationship thereby creating a triad of concepts for solving Problem P. In a nutshell, by postponing the limiting process to the last step, the CMP approach becomes remarkably powerful in much the same way as Pontryagin's Hamiltonian is significantly more powerful than Hamiltion's Hamiltonian[32] given that the only difference between them is a commutative operation. Although the CMP is essentially modern in its outlook, its roots can be traced back to the works of Euler, Leibnitz and Bernoulli.[26] Since the CMP blurs the distinction between the traditional notions of "analytic" and "numerical" solutions, it is instructive to examine the meaning of an apparently simple word, "solution," in order to provide a proper perspective to the CMP approach.

2. A Perspective on Analytic Solutions

We advance the notion that a solution is essentially a special type of approximation. Theoretical and practical considerations support this notion. Consider for example three solutions to a problem written as, $x_1 = \pi$, $x_2 = 3.14159$, and $x_3 \approx \pi$. Suppose that we regard x_1 as the exact solution. It remains a symbolic solution until it is computed. Considering that research on the computation of π has spanned 2000 years and continues to this day,[1] we regard $x_1 = \pi$ to be an approximate solution masquerading as an exact solution. Thus, there is no difference between the "exact" solution, x_1, and the "approximate" solution, x_3, if the precise nature of the approximation is clarified. In fact, x_3 is a more honest representation of the solution. In this spirit x_2 is the most useful solution and makes the exact and approximate solutions equivalent.

Now consider the field of real numbers, $\mathbb{R}$. Since $\mu(\mathbb{Q}) = 0$ where μ is the Lebesgue measure, and $\mathbb{Q}$ is the set of rational numbers, we paraphrase Cantor's famous theorem that $\mathbb{Q}$ is denumerable as "almost all numbers are irrational." Consequently, by the arguments of the preceding paragraph,

*Professor, Department of Mechanical and Astronautical Engineering, Naval Postgraduate School.
Presented at Princeton University, Princeton, NJ.

we must conclude that almost all solutions are approximate. This philosophy is further strengthened in considering physical problems. For example, the purportedly exact solution to Galileo's hanging-chain problem[21] is the equation for the catenary,

$$y(x) = k\cosh\left(\frac{x - C_1}{k}\right) + C_2 \tag{1}$$

where $k = T_0/\rho g$, T_0 is the unknown tension at 0 (see Fig. 1), ρ is the mass density of the cable assumed to be uniform, g

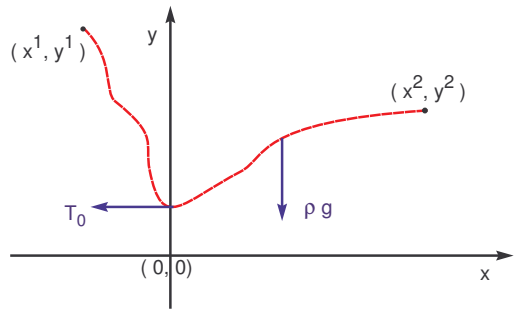

Fig. 1 Galileo's hanging-chain problem.

is the acceleration due to gravity, and C_1 and C_2 are constants of integration. The three unknowns, T_0 (and hence k), C_1 and C_2 are determined from the given length of the cable, and the coordinates of the end points, (x^1, y^1) and (x^2, y^2). According to conventional wisdom, solutions given in an analytical form are exact solutions. Hence, (1) qualifies as an "exact" solution; however, note that this equation is truly an intermediate step in solving the catenary problem. That is, a solution is expressed in terms of a presumably simpler problem of solving three nonlinear equations simultaneously; hence, (1) is, in effect, an intermediate step in formulating a function, $\mathcal{F} : \mathbb{R}^3 \to \mathbb{R}^3$, that needs to be solved for its zeros. As Nahin[21] notes, finding these parameters is a formidable task.

Note also that the hyperbolic cosine function can be defined in terms of the limit of an infinite power series. Ignoring the details of this series, (1) is equivalent to,

$$y(x) = \sum_{i=0}^{\infty} \beta_i(x) \tag{2}$$

where $\beta_i(x)$ are polynomials. The computation of (1) is almost always via an appropriate version of (2) (e.g. computations using an electronic calculator*) while (2) is computed by truncation of the series based on some acceptable upper bound on the error. In other words, (1) is just as accurate as

$$y(x) \approx k\cosh((x - C_1)/k) + C_2$$

or

$$y(x) \approx \sum_{i=0}^{N} \beta_i(x), \quad N < \infty$$

with a proper clarification of the nature of the approximation. Thus, even in the extreme case of perfect knowledge of the constants in (1), the evaluation of the function itself is an approximation. Thus, (1) is approximated at least twice: once in the evaluation of the hyperbolic function, and another in the algorithm for finding the zeros of $\mathcal{F}$ towards the computation of the unknown constants. These two layers of approximations are in addition to the implicit approximation of computations over real numbers (Cantor's theorem).

Note that all of the preceding arguments hold even in the absence of a digital computer. The introduction of a digital computer introduces a fourth layer of approximation due to the fundamental fact that digital computers operate over finite (Galois) fields. Finally, the fifth and sixth layers of approximation stem from two other perfect situations:

1. Exact knowledge of the system parameters (e.g. $\rho, x^1, \ldots$), and
2. Ideal system properties (e.g.: uniform ρ).

These two layers of approximation stem from converting a physical problem to a mathematical one. That is even if there was such a thing as an exact solution to a mathematical problem, it would almost always represent an approximate solution to a physical problem. These facts are particularly inescapable in astrodynamics.

A final point worth noting is the connection between approximations and the notion of feedback in control theory. Almost the entire theory of feedback control is based on the presumption that exact models for systems cannot be obtained. If exact solutions to exact problems were possible, feedback would be unnecessary. That approximations are inherent and fundamental essentially underpins the entire field of control theory.

3. The $\mathcal{LSA}(P)$ Philosophy

In solving Problem P, equations such as (1) are obtained by the $\mathcal{SLA}(P)$ approach. That is, equations are obtained after some limiting process (originally, the calculus of variations in Galileo's problem and standard optimal control theory of Bellman or Pontryagin in the current context). The solution is then expressed in terms of presumably simpler problems (e.g. finding the zeros of coupled nonlinear equations), and a final solution is described by way of approximation theory (hidden numerical analysis e.g. in expanding $y(x)$ in terms of truncating an infinite series of some other easily computable functions, β_i). Thus, the $\mathcal{SLA}(P)$ approach can be categorized as an equation-centric philosophy. In contrast, the $\mathcal{LSA}(P)$ approach is solution centric: one begins by directly writing the solution as (2) and using a finite expansion based on a desired level of accuracy. Thus the limiting process is performed for theoretical necessity and only once. In contrast, in the $\mathcal{SLA}(P)$ approach, the limiting process is carried out twice: once to completion to arrive at equations and then a second time for "numerical purposes" to solve the equations. If the numerics are difficult, the problem is declared difficult. In the remainder of this paper we show that by commuting the limiting process, that is, using the $\mathcal{LSA}(P)$ approach, difficult problems can be rendered easy. This gives credence to the notion that it was the *path* to problem solving (i.e. the $\mathcal{SLA}(P)$ path) that generated hard subproblems while Problem P itself might have been easy if the right path (i.e. the $\mathcal{LSA}(P)$ path) was chosen. Thus, for example, while (1) does not hold when ρ is not uniform, the $\mathcal{LSA}(P)$ process implicit in (2) holds under more general conditions. Thus, in using the word, "exact" in a relative sense, it can be said that the $\mathcal{LSA}(P)$ philosophy favors approximate solutions to "exact problems" while the $\mathcal{SLA}(P)$ philosophy favors "exact solutions" to approximate problems.

We hasten to note that we are not advocating the solution-centric philosophy over the equation-centric approach for all problems; rather, a better analogy is from mechanics in terms of the distinction between Newton's method and Lagrange's method for obtaining equations of motion. If the endgame of dynamics were to arrive at equations of motion (not solutions to them), Newton's method while attractive and visual (in terms of free-body diagrams) is highly cumbersome for complex systems while Lagrange's method is a "mechanized" easy process whose effectiveness is more pronounced with an increase in the complexity of the dynamical system. Analogously, in the catenary example, if the problem was posed with a nonuniform mass density for the chain, the $\mathcal{SLA}(P)$ approach would be severely limited.

*Most of these computations are based on the Coordinate Rotation Digital Computer (CORDIC) algorithm.[2]

As will be apparent later, the $\mathcal{LSA}(P)$ approach poses no major obstacles.

The rest of the paper is devoted to providing the technical justification for the $\mathcal{LSA}(P)$ approach leading to the CMP and its relationship to the Bellman and Pontryagin frameworks. The CMP completes and modernizes a triad of concepts for solving optimal control problems. As a matter of quick reference and completeness, we first review the issues in solving optimal control problems by the Bellman and Pontryagin methods.

4. Issues in the Bellman and Pontryagin Methods

Consider the following optimal control problem:

$$(\mathsf{P}) \begin{cases} \text{Minimize} & J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_0, t_f] = \\ & E(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) + \displaystyle\int_{t_0}^{t_f} F(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \\ \text{Subject to} & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ & \mathbf{u}(t) \in \mathbb{U}(t) \\ & (\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) \in \mathbb{E} \end{cases}$$

where $\mathbb{E}$ is some given endpoint set, $\mathbb{U}(t) \subseteq \mathbb{R}^{N_u}$ is the control space (a set valued time-dependent map), $E : \mathbb{E} \to \mathbb{R}$, $F : \mathbb{X} \times \mathbb{U} \times \mathbb{R} \to \mathbb{R}$, $\mathbf{f} : \mathbb{X} \times \mathbb{U} \times \mathbb{R} \to \mathbb{R}^{N_x}$, and $[t_0, t_f] \subset \mathbb{R}$. Typically, the state space, $\mathbb{X}$, is an open set in $\mathbb{R}^{N_x}$ but state constraints of the form, $\mathbf{x}(t) \in \mathbb{X} \subset \mathbb{R}^{N_x}$ can also be added to the problem. In order to limit the scope of the discussion, we focus on Problem P as formulated above. The problem is to find a state-control function pair, $\{\mathbf{x}(\cdot), \mathbf{u}(\cdot)\}$ (in some appropriate function space), and possibly the boundary times t_0 and t_f that minimize J. Whether or not one approaches solving Problem P from the Bellman or Pontryagin frameworks, both methods require a solution to the Hamiltonian Minimization Condition,

$$(\mathsf{HMC}) \begin{cases} \underset{\mathbf{u}}{\text{Minimize}} & H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}, t) \\ \text{Subject to} & \mathbf{u} \in \mathbb{U}(t) \end{cases}$$

where H is the *control* Hamiltonian defined as,

$$H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}, t) := F(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$$

In the framework of the Minimum Principle, $\boldsymbol{\lambda} \in \mathbb{R}^{N_x}$ is the costate where $t \mapsto \boldsymbol{\lambda}$ satisfies the adjoint equations while in Bellman's dynamic programming framework, $\boldsymbol{\lambda} = \partial_{\mathbf{x}} \varphi$ where, $\varphi : \mathbb{R} \times \mathbb{R}^{N_x} \to \mathbb{R}$, satisfies the Hamilton-Jacobi partial differential equation,[8]

$$\mathcal{H}\left(\partial_{\mathbf{x}} \varphi(t, \mathbf{x}), \mathbf{x}, t\right) + \partial_t \varphi(t, \mathbf{x}) = 0 \tag{3}$$

where $\mathcal{H} : \mathbb{R}^{N_x} \times \mathbb{R}^{N_x} \times \mathbb{R} \to \mathbb{R}$ is the *lower* Hamiltonian defined as,[8]

$$\mathcal{H}\left(\boldsymbol{\lambda}, \mathbf{x}, t\right) := \min_{\mathbf{u} \in \mathbb{U}} H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}, t)$$

and $\partial_{\mathbf{x}} \varphi(t, \mathbf{x})$ is the (Fréchet) derivative of φ with respect to $\mathbf{x}$ if $\varphi \in C^1$ exits; otherwise, one must invoke nonsmooth formalisms for $\partial_{\mathbf{x}} \varphi(t, \mathbf{x})$ (for example, the proximal subdifferential[8]). In general, Problem HMC cannot be solved for $\mathbf{u}$ in closed form; in fact, in many real-world applications, Problem HMC is solved quite simply and efficiently through nonlinear programming (NLP) techniques. To illustrate this key point, consider the simple case when $\mathbf{f}$ is differentiable with respect to $\mathbf{u}$, and $\mathbb{U}$ is given in terms of function inequalities as follows,

$$\mathbb{U} := \left\{ \mathbf{u} \in \mathbb{R}^{N_u} : \mathbf{h}^L \leq \mathbf{h}(\mathbf{u}) \leq \mathbf{h}^U \right\}$$

where $\mathbf{h} : \mathbb{R}^{N_u} \to \mathbb{R}^{N_h}$ is a continuously differentiable function and $\mathbf{h}^L, \mathbf{h}^U \in \mathbb{R}^{N_h}$ are the lower and upper bounds on the values of the function $\mathbf{h}$ respectively. Then, Problem HMC is a an NLP. In general, an NLP does not have a closed form solution. Thus, barring a very limited number of cases (e.g. an unconstrained Hamiltonian quadratic in $\mathbf{u}$), it is quite difficult to obtain the map,

$$(\boldsymbol{\lambda}, \mathbf{x}) \mapsto \mathbf{u}^* = arg \min_{u \in \mathbb{U}} H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{u})$$

Thus, in the absence of a major breakthrough in closed form solutions for NLPs, it is clear that Problem HMC poses the first and important bottleneck in seeking closed-form solutions to general optimal control problems. Consequently, many astrodynamics problems are formulated with simplified Hamiltonians and control spaces ($\mathbb{U}$) so that $\mathbf{u}^*$ can be formally obtained in closed form (i.e. in seeking "exact" solutions to approximate problems). In crossing this bottleneck through substantial simplification, the HJB now produces a partial differential equation (PDE) in N_x variables. This generates at least two additional well-known impediments. The first problem is that the HJB may not have a solution even when an optimal solution to Problem P exists. That is, a differentiable function, φ, that solves the HJB may not exist. As is well documented,[2,3] this absence of a differentiable function is more common than rare. This difficulty was overcome in the 1980s by expanding the notion of differentiability to "non-differentiable" functions by way of two different approaches, namely the viscosity notion pioneered by Crandall and Lions[4] and the notion of generalized derivatives (e.g proximal subdifferential) pioneered by Clarke.[7] Despite these theoretical breakthrough, a second and practical problem related to the dimension of N_x remains. That is, even with an extension of the notion of differentiability to nonsmooth functions, solving a PDE in more than two or three variables is such a difficult problem that Bellman himself referred to it as the "curse of dimensionality." In other words, the Hamilton-Jacobi road map is beset with significant problems in almost all steps towards a solution. Note that the HJB equations can be defined in terms of "Bellman's $\mathcal{LA}(P)$" with bottlenecks in the $\mathcal{SL}$ operations of $\mathcal{SLA}(P)$.

In the alternative framework of Pontryagin, the necessary conditions for Problem HMC can be stipulated by way of the Karush-Kuhn-Tucker (KKT) conditions. These algebraic equations, together with the adjoint equations provide the necessary conditions in dual space. Thus, the totality of necessary conditions can be summarized in terms of a nonlinear boundary value problem (BVP) that solves Problem P. The BVP is a differential-algebraic problem with boundary conditions specified in terms of the transversality conditions. This problem is "Pontryagin's $\mathcal{LA}(P)$." It is well-known that solving a differential-algebraic BVP is a difficult problem.[3] A significant additional difficulty that is tagged on to this BVP is that it constitutes a Hamiltonian system in $2N_x$ dimensions. The well-known symplectic structure of a Hamiltonian system implies that the associated BVP is unstable with respect to propagation in either the forward or backward direction. Hence, the problem of finding the initial conditions on the costates is "chaotic" in the sense that small perturbations produce wild trajectories in the primal-dual space.[6] Although this problem can be alleviated by the use of symplectic integrators, note that symplectic integration requires some *a priori* qualitative knowledge of the behavior of the Hamiltonian system. Thus, whereas in Hamiltonian dynamics one perform symplectic integration in the position-velocity "physical space" (primal space), in optimal control, the symplecticity is in the more abstract primal-dual space.

It is thus apparent that the equation-centric methods (i.e. $\mathcal{SLA}(P)$ philosophy) of either Bellman or Pontryagin lead to difficult subproblems. We will now show that commuting the last two operations renders difficult problems easy in much the same way as Pontryagin's Hamiltonian, H, obviates the difficulties of Hamilton's Hamiltonian, $\mathcal{H}$, by a

single stroke of commutation[32] (in differentiation and minimization operations). A key idea that facilitates the notion of an $\mathcal{S}-\mathcal{L}$ commute is the concept of inexact problems.

5. The Inexaction Key

The $\mathcal{LSA}(P)$ philosophy requires that a solution to an approximate problem $\mathcal{A}(P)$ be first obtained thereby postponing the limiting operation, $\mathcal{L}$, to the last step. This concept can be traced back to Bernoulli who obtained a solution to the Brachistochrone problem by approximating it to a problem where Fermat's principle could be applied.[21,26] In generalizing this concept, we use approximation theory to construct $\mathcal{A}(P)$ and algorithms to generate $\mathcal{SA}(P)$. Unlike the days of Bernoulli where it was essential to obtain $\mathcal{SA}(P)$ "by hand," we exploit a modern computer to obtain this in "real-time." In principle, it is not necessary to utilize a computer, and one could obtain solutions to $\mathcal{A}(P)$ by hand as well. An example of this process is shown below to illustrate this point. In this case, this process is equivalent to seeking "closed-form solutions" as in the $\mathcal{S}$-stage of the $\mathcal{SLA}(P)$ framework (Bellman or Pontryagin). Typically, the $\mathcal{S}$-stage involves an algorithm for non-trivial problems. The last stage of taking limits in $\mathcal{LSA}(P)$ is crucial to this process in order to demonstrate that we have indeed solved the correct Problem P. In fact, if this convergence test is not performed, the $\mathcal{LSA}(P)$ process could easily generate spurious solutions. Consequently, we note that a necessary condition to ensure that the correct problem is being solved is that the results of the $\mathcal{LSA}(P)$ and $\mathcal{SLA}(P)$ approaches be identical; that is we require that $\mathcal{L}$ and $\mathcal{S}$ be commutative. By definition, this simply means that the commutation hold for an "exact" solution. In order to allow the exact solution to be contained within the set of approximate solutions obtained at the $\mathcal{SA}(P)$-stage, we require that *approximate problems be solved "inexactly"*. This is in sharp contradistinction to the conventional wisdom of seeking exact solutions to approximate problems. To illustrate these key points, consider the simple univariate initial value problem:

$$(\mathsf{Q})\begin{cases} \dot{x}(t) = & ax(t) \\ x(0) = & 1 \end{cases} \tag{4}$$

where a is a given real number. Although this is not an optimal control problem, it supports a simple illustration of the foregoing statements. We assume that we can write a solution to Problem Q as,

$$x^{\infty}(t) = \sum_{n=0}^{\infty} \alpha_n(t) \tag{5}$$

Since we already know the exact solution to be given symbolically as e^{at}, it is obvious that one possible representation for (5) is given by the power series,

$$x_1^{\infty}(t) = \sum_{n=0}^{\infty} \frac{(at)^n}{n!} \tag{6}$$

From the conclusions of Sec. 2 that it is impossible to compute the exact solution exactly, we seek to compute a partial sum,

$$x^{N}(t) = \sum_{n=0}^{N} \alpha_n(t)$$

so that $d(x^{\infty}(\cdot), x^{N}(\cdot)) \leq \varepsilon$ where d is some appropriate metric and $\varepsilon \geq 0$ is a tolerance in error. Note that for all finite N we have

$$\dot{x}_1^N(t) \neq a x_1^N(t)$$

but the initial condition, $x(0) = 1$, is satisfied exactly. Now suppose that we did not know the exact solution to Problem Q, as is the case in general. One of the simplest techniques to solve the problem (i.e. generate the partial sums) are discretization methods which can be viewed as approximations in the time-domain in contrast to frequency-domain methods (e.g. generalized Fourier expansion). For example, the standard explicit Euler method seeks to find $\alpha_n(t)$ according to the following procedure: construct $\mathcal{A}(Q) = Q_h$; that is, a family of algebraic problems parameterized by, h:

$$(\mathsf{Q}_h)\begin{cases} x_{k+1} = & x_k + hax_k \qquad k = 0, 1, \ldots \\ x_0 = & 1 \end{cases} \tag{7}$$

The equalities in Problem Q_h explicitly require that the equations be solved exactly; that it, in the standard explicit Euler method, we seek to find *exact solutions to the approximate problem*, Q_h, leading to the result,

$$x_N = x_{N-1}(1+ha) = x_{N-2}(1+ha)^2 = \ldots = (1+ha)^N \tag{8}$$

Now, given any point t^*, we can always find a pair (N, h) such that $t^* = Nh$, so that we can write,

$$x^N(t^*) = (1 + at^*/N)^N = \sum_{n=0}^{N} \frac{N!}{n!(N-n)!N^n}(at^*)^n \tag{9}$$

where the last term in (9) is the result of a binomial expansion. In any event, we have $\lim_{N\to\infty} x^N(t^*) = e^{at^*}$. This last step of convergence completes a version of the $\mathcal{LSA}(Q)$ operations in a conventional sense that implicitly require that $\mathcal{L}$ and $\mathcal{S}$ be commutable in the limit. In recognizing that an exact solution is essentially a theoretical limit and largely both impossible and unnecessary (see Sec. 2), we propose $\mathcal{L}$ and $\mathcal{S}$ to commute for approximate solutions as well; hence, we abandon the notion of pursuing exact solutions to approximate problems as implied by the equalities in (7). In favor of seeking inexact (approximate) solutions to the approximate problem, Q_h, we write,

$$(\mathsf{Q}_h)\begin{cases} x_{k+1} \approx & x_k + hax_k \qquad k = 0, 1, \ldots \\ x_0 \approx & 1 \end{cases} \tag{10}$$

where we have abused notation in referring to (10) as also Problem Q_h. The precise nature of the inexactness indicated in (10) will be apparent shortly. Because of the inexactness in (10), the recursive step of (8) cannot be carried out as it is unclear how to generate x_{k+1} from x_k. As a result of this fundamental shift in viewing approximations, we distinguish (7) from (10) by referring to the former as the *exact Euler method* while the latter is referred to as the *inexact Euler method*. Naturally, all available approximation methods can be distinguished in a similar manner and thus this terminology is applicable to non-Eulerian methods as well. The most natural way to formulate a solution via inexact methods is to treat (10) as a problem with an unknown global variable, X, defined as,

$$X := \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{pmatrix} \in \mathbb{R}^{N+1}$$

where $Nh = t^*$ and t^* is any given point as before. Then (10) is recast as system of *inexact* equations,

$$\underbrace{\begin{pmatrix} -1-ah & 1 & 0 & \cdots & 0 \\ 0 & -1-ah & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1-ah & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix}}_{(N+1)\times(N+1)} \underbrace{\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{pmatrix}}_{X} \approx \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}}_{(N+1)\times 1} \tag{11}$$

In order to demonstrate how this simple idea makes an enormous difference, consider the exact solution, $x(kh) = e^{akh}$ as candidate solutions for the components of X with $x_k = x(kh)$. The initial condition, $x(0) = 1$ (for $k = 0$) is satisfied exactly by (11) via the last row; hence it is satisfied approximately as well. All other rows satisfy,

$$-1 - ah + e^{ah} \approx 0$$

meaning that for any given tolerance, $\varepsilon > 0$, we can find some h such that,

$$\left|-1 - ah + e^{ah}\right| \leq \varepsilon$$

Thus, the exact solution is a candidate solution to (11) and hence to (10). Needless to say that since

$$-1 - ah + e^{ah} \neq 0$$

the exact solution is not a solution for (7) which is based on the exact Euler method. We thus end up with a simple rule: *Solve approximate problems inexactly; or conversely, do not solve approximate problems exactly.* This apparent luxury for initial value problems turns out to be a necessity for solving optimal control problems via the covector mapping principle (discussed in the next section). Our notion of inexactness is a generalization of the relaxation schemes proposed in the 1960s. The main issue in the 1960s was on the validity of the Minimum Principle (specifically, Problem HMC), for discrete-time systems; see, for example [15] and the upcoming book [20]. The key issues centered around convexity and discrete versions of the Minimum Principle. The workhorse for discretization was the *exact* Euler method and relaxations (inexactness) essentially centered around the boundary conditions (and not on the differential equations as required by inexact methods). It was believed that the ideas based on (exact) Euler methods would carry over to other types of (exact) discretizations with the details differing only in the rates of convergence. That this is not true is a recent discovery,[10,14] leading to a new revival of discrete methods. In addition, inexact methods desensitize the sensitivity of the Hamiltonian system which arises chiefly as a result of seeking exactions to symplectic boundary value problems. These new ideas suggest that some of the conjectures on higher-order methods postulated in 1960s turn out to require significant modifications. The clarity of concepts borne out by a revival of approximation theory, advances in computing technology and new methods of approximation is encapsulated by the covector mapping principle.

6. The Covector Mapping Principle

The covector mapping principle (CMP) is centered around the key idea of commuting $\mathcal{L}$ and $\mathcal{S}$ operations in the approximation spaces *as well as their duals.* This notion has a domino effect. As alluded to in Sec. 5, a key idea that permits such a commutation is the concept of inexact problems. Inexactness with respect to discretization implies that there is no distinction between implicit and explicit methods. Typically, there is no unique solution; rather, each inexact problem generates a possibly non-singleton set of solutions. Under appropriate conditions, a subset of solutions are close to one another (in some appropriate metric). In addition, a theoretical exact solution is an element of this solution set.

Based on these broad ideas, the CMP approach begins in much the same manner as the Bellman and Pontryagin approaches by assuming the existence of an optimal solution to Problem P. Rather than obtain conditions (i.e. equations) that the solution must satisfy, we conceive of an optimal solution as the limiting process of a sequence. For example, suppose that the optimal system trajectory, denoted as $[t_0^\infty, t_f^\infty] \mapsto \{\mathbf{x}^\infty, \mathbf{u}^\infty\}$, can be written as an infinite series,

$$\mathbf{x}^\infty(t) = \sum_{k=0}^{\infty} \boldsymbol{\alpha}_k(t) \qquad \mathbf{u}^\infty(t) = \sum_{k=0}^{\infty} \boldsymbol{\beta}_k(t) \tag{12}$$

The partial sums defined as,

$$\mathbf{x}^N(t) = \sum_{k=0}^{N} \boldsymbol{\alpha}_k(t) \qquad \mathbf{u}^N(t) = \sum_{k=0}^{N} \boldsymbol{\beta}_k(t) \tag{13}$$

form the requisite sequence. A conceptually simple way to conceive of $\left\{\mathbf{x}^N(\cdot), \mathbf{u}^N(\cdot)\right\}$ is by way of Weierstrass' Approximation Theorem which guarantees a sequence of polynomials converging to any given continuous function on a finite (compact) time-interval, $[t_0^\infty, t_f^\infty]$. Note that $\left\{\mathbf{x}^N(\cdot), \mathbf{u}^N(\cdot)\right\}$ in (13) are not necessarily polynomial representations. In any event, $\left\{\mathbf{x}^N(\cdot), \mathbf{u}^N(\cdot)\right\}$ are, by definition, approximate solutions to Problem P with errors given by some appropriate metric. From the notion of inexact problems introduced in Sec. 5, it is apparent that we should be able to construct Problems P^N such that the exact solution to Problem P is also a solution to Problem P^N. Given that the notion of inexactness is intuitive but somewhat vague in mathematical precision, we use the word "principle" in the same spirit. That is, a principle is a generator of theorems and is by nature imprecise with broad reach while a theorem makes it precise with a narrower reach. In this context we formulate the CMP as follows:

Proposition 1 (Covector Mapping Principle) *Let $[t_0^\infty, t_f^\infty] \mapsto \{\mathbf{x}^\infty, \mathbf{u}^\infty\}$ be an exact solution to a given optimal control problem P. Then, under appropriate conditions, there exists a sequence of inexact problems $\left\{P^N\right\}_{N=0}^{\infty}$ such that,*

a) $[t_0^\infty, t_f^\infty] \mapsto \{\mathbf{x}^\infty, \mathbf{u}^\infty\}$ is a solution for Problem P^N for all finite N, and

b) Problem P^∞ = Problem P,

In addition for all such sequences, there exists

c) time intervals $\left\{[t_0^N, t_f^N]\right\}_{N=0}^{\infty}$,

d) system trajectories, $\left\{t \mapsto (\mathbf{x}^N, \mathbf{u}^N)\right\}_{N=0}^{\infty}$,

d) (covector) functions $\left\{t \mapsto \boldsymbol{\lambda}^N\right\}_{N=0}^{\infty}$, and

e) mappings $\left\{\Gamma^N : \mathbb{R}_{N_x} \to \mathbb{R}_{N_x}\right\}_{N=0}^{\infty}$,

such that for all finite N,

1. *$[t_0^N, t_f^N] \mapsto \left\{\mathbf{x}^N, \mathbf{u}^N, \boldsymbol{\lambda}^N\right\}$ satisfies the optimality conditions for Problem P inexactly,*
2. *$t \mapsto \left\{\mathbf{x}^N, \mathbf{u}^N, \Gamma^N(\boldsymbol{\lambda}^N)\right\}$ satisfies the inexact optimality conditions for Problem P^N,*

and

$$\lim_{N\to\infty} \left\{t \mapsto \Gamma^N(\boldsymbol{\lambda}^N)\right\} = \lim_{N\to\infty} \left\{t \mapsto \boldsymbol{\lambda}^N\right\} = \{t \mapsto \boldsymbol{\lambda}^\infty\}$$

where $[t_0^\infty, t_f^\infty] \mapsto \boldsymbol{\lambda}^\infty$ together with $[t_0^\infty, t_f^\infty] \mapsto \{\mathbf{x}^\infty, \mathbf{u}^\infty\}$ satisfy the exact optimality conditions for Problem P.

All that it needed to make the CMP precise are unambiguous definitions of "appropriate conditions", "inexact optimality conditions," and other terms in the statement of Proposition 1. In adding this precision to the principle, we end up with covector mapping theorems. A schematic of the ideas associated with the CMP is shown in Fig. 2. The superscript λ on problems in Fig. 2 refers to problems generated by applying the $\mathcal{SLA}(P)$ approach. Thus, for example, Problem P^λ refers to the boundary value problem that is coupled with Problem HMC when the Pontryagin framework is applied (Pontryagin's $\mathcal{LA}(P)$ in Sec. 4); in the case of the Bellman framework, Problem P^λ is the HJB equation (Bellman's $\mathcal{LA}(P)$ in Sec. 4). For the purpose

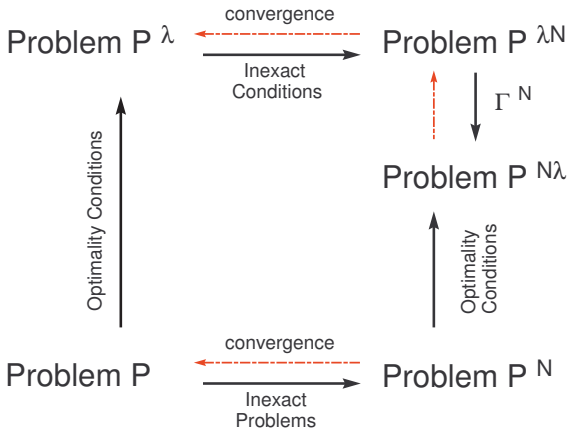

Fig. 2 The Covector Mapping Principle.

of brevity, we discuss only the optimality conditions associated with the Pontryagin style conditions, while simply noting that the discussions apply to the Bellman framework as well. Furthermore, we can think of the superscript N as resulting from discretizations. In this case, Problem P^N is an NLP (if $\mathbb{U}$ is a continuous set) and Problem $P^{N\lambda}$ is the mixed complementarity problem resulting from an application of the KKT theorem. All of the problems obtained in the right of Fig. 2 are inexact problems in much the same way as (10) is distinguished from (7).

Before illustrating the CMP process, note the following:

Remark 1.1 The CMP does not say that solving a sequence of Problems P^N will generate a sequence of solutions for Problem P even if P^N is designed such that $\lim_{N\to\infty} P^N = P$. Although this is a highly desirable result, it is a convergence type theorem that needs to be developed specifically for specific schemes that generate P^N.

Remark 1.2 When P^N is an NLP, note that the CMP does not say that the multipliers of the NLP converge to the multipliers of P. Rather, the CMP states that there exists multipliers for the NLP that converge to the multipliers of P. These convergent sequence of multipliers are given by the mappings Γ^N. The subscripts on $\mathbb{R}$ in statement e) of Proposition 1 essentially emphasize that these are transformations in the dual space.

To illustrate a process in how the CMP may be applied, consider an optimal solution, $[t_0, t_f] \mapsto \{\mathbf{x}, \mathbf{u}\}$, of Problem P. We now construct $[t_0, t_f] \mapsto \{\mathbf{x}^N, \mathbf{u}^N\}$ by interpolating the optimal solution over a grid, $\pi^N = [t_0, t_1, \ldots, t_N = t_f]$,

$$\mathbf{x}^N(t) = \sum_{i=0}^{N} \mathbf{x}(t_k)\psi_k(t), \quad \mathbf{u}^N(t) = \sum_{i=0}^{N} \mathbf{u}(t_k)\psi_k(t) \tag{14}$$

where $\psi_k(t)$ are interpolating functions; i.e, functions that satisfy the Kronecker delta condition, $\psi_k(t_j) = \delta_{kj}$. In order to construct Problems P^N, we define,

$$\widetilde{\mathbf{x}}^N(t) = \sum_{i=0}^{N} \mathbf{x}_k\psi_k(t), \quad \widetilde{\mathbf{u}}^N(t) = \sum_{i=0}^{N} \mathbf{u}_k\psi_k(t) \tag{15}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ are unknowns. It is not too difficult to see by the arguments outlined in Sec. 5 that Problems P^N can be constructed by "substituting" $\widetilde{\mathbf{x}}^N(t)$ and $\widetilde{\mathbf{u}}^N(t)$ in Problem P so that the approximate solution $\{\mathbf{x}^N(t), \mathbf{u}^N(t)\}$ and the exact solution $\{\mathbf{x}(t), \mathbf{u}(t)\}$ are both solutions to Problem P^N. One of the simplest ways to construct these problems is by way of pseudospectral (PS) methods[11,29] where $\psi_k(t)$ are chosen to be Lagrange interpolating polynomials with π^N based on some appropriately chosen "quadrature" points. This choice of points guarantees $\lim_{N\to\infty} \mathbf{x}^N(t) = \mathbf{x}(t)$. Such methods of constructing P^N can be described as time-domain methods. In the frequency-domain methods, one begins by approximating $\{\mathbf{x}(t), \mathbf{u}(t)\}$ by

$$\widetilde{\mathbf{x}}^N(t) = \sum_{i=0}^{N} \mathbf{a}_k\Omega_k(t), \quad \widetilde{\mathbf{u}}^N(t) = \sum_{i=0}^{N} \mathbf{b}_k\Omega_k(t) \tag{16}$$

where $\Omega_k(t)$ are basis functions for an expansion of $\{\mathbf{x}(t), \mathbf{u}(t)\}$. Typically, these are chosen to be orthogonal functions so that an orthogonal "projection" of (16) into Problem P generates P^N with unknowns coefficients.

Other variants of Problems P^N can be generated using (15) or (16). For example, instead of finding a pair, $\{\mathbf{x}(\cdot), \mathbf{u}(\cdot)\}$, as implied in the definition of Problem P, one could *a priori* choose $\mathbf{x}(t)$ as the solution to the differential equation,

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

in which case $\widetilde{\mathbf{x}}^N(t)$ is implicitly determined from,

$$\dot{\widetilde{\mathbf{x}}} = \mathbf{f}(\widetilde{\mathbf{x}}, \widetilde{\mathbf{u}}^N(t)), \quad \widetilde{\mathbf{x}}(t_0) = \mathbf{x}_0$$

The difficulty in such an approach is an easy description of generating inexact problems since as already noted in Sec. 5, the inexaction step is key to generating a proper family of solutions. In any case, it is clear that by solving an appropriate Problem P^N, we can obtain solutions to any desired precision (assuming convergence; see Fig. 2). This is the $\mathcal{LSA}(P)$ approach in a nutshell. Obviously, it is essential to fully exploit tools from approximation theory and functional analysis to use this method effectively. In tying this solution to the $\mathcal{SLA}(P)$ approach, we can devise intuitive verification methods so that all the tools of the traditional methods (Bellman and Pontryagin) can be fully exploited; see for example, [25]. To this end, we restrict our attention to the Pontryagin framework in explaining P^λ. From the Minimum Principle it follows that there exists a covector function, $[t_0, t_f] \mapsto \boldsymbol{\lambda}$, satisfying the adjoint equation. Consequently, for the interpolation method of (14), we can construct,

$$\boldsymbol{\lambda}^N(t) = \sum_{i=0}^{N} \boldsymbol{\lambda}(t_k)\psi_k(t), \quad \widetilde{\boldsymbol{\lambda}}^N(t) = \sum_{i=0}^{N} \boldsymbol{\lambda}_k\psi_k(t) \tag{17}$$

Since $\boldsymbol{\lambda}^N(t)$ is an approximation to $\boldsymbol{\lambda}(t)$, it is apparent that $\boldsymbol{\lambda}^N(t)$ satisfies the adjoint equation inexactly and consequently can be framed as part of the inexact conditions for Problem P (see also the top horizontal arrow in Fig. 2). The last part of the statement of the CMP referring to the "inexact optimality conditions for Problem P^N" requires further explanation since Problem P^N is an inexact problem (compare (10) and (7)). The inexact conditions are obtained by first writing the KKT conditions for the exactions and then relaxing these conditions to generate inexact optimality conditions for the inexact problem. Erroneous results will be obtained if these operations are commuted! This is yet another example of how a proper commutation of operations yields correct results. In any case, a discrete-time state trajectory, $t \mapsto \mathbf{x}^N$, $t \in \pi^N$, is part of the solution set to Problem P^N. By the same token, the continuous-time state trajectory, $t \mapsto \mathbf{x}^N$, $t \in [t_0, t_f]$, can also be viewed as part of the solution set (under appropriate conditions).

One of the most important aspects of the CMP is the existence of mappings Γ^N and the manner in which it facilitates analysis, verification and feedback control.

7. The Covector Map, Γ^N

In a nutshell, Γ^N is a transformation of the multipliers (in the Pontrygain framework) from Problem $P^{\lambda N}$ to Problem $P^{N\lambda}$ (see Fig. 2). Since Problem P^N is an abstract

representation of (inexact) direct methods while Problem $P^{\lambda N}$ corresponds to indirect methods, it is apparent that what is really desirable is an inverse map, $(\Gamma^N)^{-1}$, as direct methods are significantly easier than indirect methods.[5] In essence, Γ^N and its inverse (if it exists) provides the connections between the $\mathcal{SLA}(P)$ and $\mathcal{LSA}(P)$ approaches by requiring that $\mathcal{S}$ and $\mathcal{L}$ commute for finite N. This commutation is facilitated by the notion of inexact methods which is based on the principle of seeking inexact solutions to approximate problems rather than the conventional wisdom of seeking exact solutions to approximate problems. This shift in philosophy also requires that we abandon the notation of a single solution even under uniqueness conditions leading to the concept of solution sets. Under uniqueness conditions these solution sets would be close to one another in some metric. In the same spirit, we entertain the notion of multiplier sets. To this end, let $\mathbb{M}^{\lambda N}$ be the multiplier set for Problem $P^{\lambda N}$ so that it forms the domain for the map, Γ^N. Likewise, we denote by $\mathbb{M}^{N\lambda}$ as the multiplier set for Problem $P^{N\lambda}$ so that we can write,

$$\Gamma^N : \mathbb{M}^{\lambda N} \to \mathbb{M}^{N\lambda} \tag{18}$$

Thus, Γ^N ensures that there exists $\boldsymbol{\lambda}^N \in \mathbb{M}^{\lambda N}$, such that $\Gamma^N(\boldsymbol{\lambda}^N) \in \mathbb{M}^{N\lambda}$. The existence of $\boldsymbol{\lambda}^N$ is a direct consequence of the existence of $t \mapsto \boldsymbol{\lambda}^\infty$ for Problem P and the notion of inexact solutions. Consequently, by the mapping principle, we are ensured that there exists $\widehat{\boldsymbol{\lambda}}^N \in \mathbb{M}^{N\lambda}$ such that

$$\widehat{\boldsymbol{\lambda}}^N \mapsto \boldsymbol{\lambda}^N \in \mathbb{M}^{\lambda N}$$

Hence there exists a map, $\widehat{\Gamma}^N$, such that $\widehat{\Gamma}^N(\widehat{\boldsymbol{\lambda}}^N) = \boldsymbol{\lambda}^N$. A procedure to obtain $\widehat{\Gamma}^N$ is conceptually simple but requires a broad set of tools from mathematical analysis. The key idea is to apply nearly the same discretization method to Problems P and P^λ to generate Problems P^N and $P^{\lambda N}$ respectively. By applying the necessary optimality conditions to Problem P^N, we can generate Problem $P^{N\lambda}$. In general, Problems $P^{N\lambda}$ and $P^{\lambda N}$ will not even resemble one another.[14] Hence it is difficult to prove the existence of Γ^N. The most obvious route to proving the existence of Γ^N is to find an explicit map. This involves seeking a coordinate transformation (of the covectors) that renders Problems $P^{N\lambda}$ similar to $P^{\lambda N}$ so that Γ^N may be determined simply by inspection. For Eulerian discretizations, it is straightforward to show that an identity map can be constructed for Γ^N if the corresponding discretization for Problem P^λ is the symplectic Euler method. Given that Eulerian methods are computationally unattractive, it is extremely desirable to obtain Γ^N (and $\widehat{\Gamma}^N$) for higher-order methods. As a result of the pioneering work of Hager,[14] we have the following result.

Proposition 2 (Hager) *There exists a class of Runge-Kutta (RK) methods for which Γ^N and $\widehat{\Gamma}^N$ are identities. This class of RK methods are symplectic for Problem P^λ and require additional conditions beyond symplecticity for convergence.*

It is worth noting that Hager's Runge-Kutta (HRK) methods can be obtained after an explicit coordinate transformation of the covectors which require that the coefficients of the "last stage" of the RK family be nonzero. Hager has shown that it is possible to have an RK method that is convergent for differential equations but divergent for optimal control. Thus, the CMP facilitates convergence results while providing an explanation for the divergence of "convergent" RK methods in terms of conditions for the existence of $\widehat{\Gamma}^N$. These conditions belong to broader set of *Closure Conditions*[28,29] which can be described as (implicit or explicit) conditions that restore information lost (in either or both the primal and dual spaces) by the discretization process. The primary source of the information loss is the "infinite" amount of information contained in the continuity of time (independent variable) including the well-known "hidden convexity." An elementary example of this is the Hamiltonian evolution equation,

$$\frac{d\mathcal{H}}{dt} = \frac{\partial H}{\partial t} \tag{19}$$

that generalizes the "first-integral" in the calculus of variations. The evolution of the Hamiltonian is based on the continuity of time which is absent in the finiteness of discretization unless additional (or closure) conditions are introduced to maintain this property in the finite dimensional space as well (i.e. the $\mathcal{L} - \mathcal{S}$ commute for finite N). The absence of the continuity of discrete time is one of the reasons why, for example, the notion of conjugate points does not exist in finite-dimensional spaces. On the other hand, if the convergence of solutions of Problem P^N with respect to N can be proved without an explicit construction of $\widehat{\Gamma}^N$ (or Γ^N), it is clear that these covector maps exist *but only in the limit*. As noted in Sec. 2, $N = \infty$ is unachievable even when functions converge to elegant analytic solutions. Thus, in seeking covector maps for finite N, closure conditions are inescapable if we desire the discrete-time solutions to maintain some of the properties of the continuous-time theoretical solutions. Thus, those continuous-time properties that are implicitly or explicitly unaccounted for in the discretization process cannot be expected to be satisfied for the discrete-time solution except possibly in a statistical sense. This is precisely the case with PS methods. In [29] it is proved that for the Legendre PS method, (19) is indeed satisfied in a statistical (average) sense when this information is not part of the closure conditions. A covector mapping theorem for the Legendre PS method is proved in [29] using all the techniques described here. In addition a convergence result is obtained in [13] while an algorithm based on the CMP is described in [30]. A version of this algorithm is implemented in the software package, DIDO,[24] while a somewhat different implementation of the Legendre PS method is available through OTIS.[22] These ideas are demonstrably superior to prior techniques in that optimal solutions can be obtained in real time[27,31,34] for a significantly large class of problems. Thus, optimal feedback control is obtained by the simple act of replacing the initial conditions, $(t_0, \mathbf{x}_0)$, by current conditions, $(t, \mathbf{x}(t))$. This technique has been successfully applied for the attitude control of NPSAT1,[31] a spacecraft being built at the Naval Postgraduate School and scheduled for launch in September 2006.

8. Example: Low Thrust SEP Mission Design

As demonstrated in the recent past (see for example [9, 12, 14, 16, 17, 19]), the CMP is a very powerful technique to solve complex mission design problems. In this section, we illustrate a few salient points in the application of the CMP to a practical, modern astrodynamics problem.

Consider the interplanetary trajectory design of a solar-electric-propelled (SEP) spacecraft. In this system, the thrust force on the spacecraft is controlled by the electric power (P_e) to the engine which in turn is obtained from the solar arrays. The power available from the solar arrays is dependent on the spacecraft state vector (specifically, the distance, r, of the spacecraft from the sun). Thus, we have a state-dependent control space, $\mathbb{U}(\mathbf{x}, t)$, where the time-dependence is also included in $\mathbb{U}$ to allow for more general models that take into account the degradation of the solar array.

In modeling the control space, we define the control variables to be P_e, the electrical power supplied to the engine and θ the thrust steering angle. Obviously, $\theta \in S^1$. In modeling the space of input power to the engine, we have,

$$P_e \in \mathbb{P} := \{P_e \geq 0 : \ P_e \leq P_{e,avail} \wedge P_{e,max}\} \tag{20}$$

Parameter	Value	Parameter	Value
d_1	1.320770	δ_1	-6.85×10^{-5}
d_2	-0.108480	δ_2	0.0
d_3	-0.116650	P^0_{BOL}	5.0 kW
d_4	0.108430		
d_5	-0.012790		

Table 1 Data for GaAs solar array.

where $P_{e,avail}$ is the available electrical power, $P_{e,max}$ is the maximum power capacity of the engine and $\wedge$ is the logical AND operation. Equation (20) mathematically models the following facts: The power available to the engine can be higher than the maximum power it was designed to handle since solar arrays are designed to provide the minimum powers at end of life (EOL). Hence, when $P_{e,avail} > P_{e,max}$, we require $P_e \leq P_{e,max}$. On the other hand, when $P_{e,avail}$ is less than $P_{e,max}$, then the maximum available electrical power to the engine is less than its maximum power capacity and hence trajectory shaping that requires more power than the available but less than the capacity is simply not feasible. The equation for the available power is given by,

$$P_{e,avail} = P_{solar}(r,t) - P_{house}(t) \tag{21}$$

where $P_{house}(t)$ is the housekeeping power at time t while $P_{solar}(r,t)$ is the solar-array power available at (solar) distance r at time t. As a result of the combined variations in the incident solar energy, the temperature of a solar cell, and other systems-level effects, the electrical power available from a practical solar array is more complicated than a simple inverse square law. A particular power model for GaAs solar array is given by[9]

$$P_{solar}(r,t) = \frac{P^0_{BOL}}{r^2}\left[\frac{d_1 + d_2/r + d_3/r^3}{1 + d_4 r + d_5 r^2}\right](1 + \delta_1 t + \delta_2 e^{\delta_3 t}) \tag{22}$$

where $d_i, i = 1,\ldots 5$ and $\delta_j, j = 1,2,3$ are "technology" constants whose numerical values are given in Table 1. By substituting (21) and (22) in (20) we arrive at an algebraic model for the control space, $\mathbb{U} = \mathbb{P} \times S^1$, where $\mathbb{P}$ must truly be written as a set-valued map, $\mathbb{P}(r,t,P_{house})$, indicating its dependence on r,t and the housekeeping power.

Now consider a preliminary planar dynamical model for a SEP spacecraft. This can be written as,

$$\ddot{\mathbf{r}} = \mathbf{g}(\mathbf{r}) + \begin{pmatrix}\sin\theta\\ \cos\theta\end{pmatrix}\frac{T(P_e)}{m} \tag{23}$$

$$\dot{m} = -\gamma(P_e) \tag{24}$$

where $\mathbf{g}(\mathbf{r})$ is the gravity model, $T(P_e)$ is the engine thrust model and $\gamma(P_e)$ is the mass flow-rate model. The control variables are P_e and θ. As a result of the discussions of the preceding paragraphs, it is clear that the vector fields (right-hand-side of the differential equations) in modern astrodynamics problems are far more complex than pure gravity fields. Thus, in considering the effects of higher-order gravity fields, it is important to first ascertain whether other effects (like SEP) might dominate the behavior of the dynamical system far more than higher-order gravity models. Furthermore, in addition to the complexity of the control space, the functions, $T(P_e)$ and $\gamma(P_e)$, for a practical SEP engine are frequently given in terms of table look up data. For the NSTAR engine, one particular model is truly non-smooth, and is given by,[9]

$$T(P_e) = \begin{cases} 0 & P_e \leq P_{e,min} \\ \sum_{i=0}^4 c_i P_e^i & P_e \geq P_{e,min}\end{cases} \tag{25}$$

$$\gamma(P_e) = \begin{cases} 0 & P_e \leq P_{e,min} \\ \sum_{i=0}^4 b_i P_e^i & P_e \geq P_{e,min}\end{cases} \tag{26}$$

where $c_i, b_i, i = 0,\ldots 4$ are coefficients of a polynomial fit to the engine data. These coefficients for the NSTAR engine

Parameter	Value	Parameter	Value
b_0	2.38	c_0	25.3
b_1	-5.08	c_1	-49.7
b_2	5.04	c_2	87.0
b_3	-2.41	c_3	-35.3
b_4	0.35	c_4	4.97
$P_{e,min}$	0.52 kW	$P_{e,max}$	2.57 kW

Table 2 Data for the NSTAR engine.

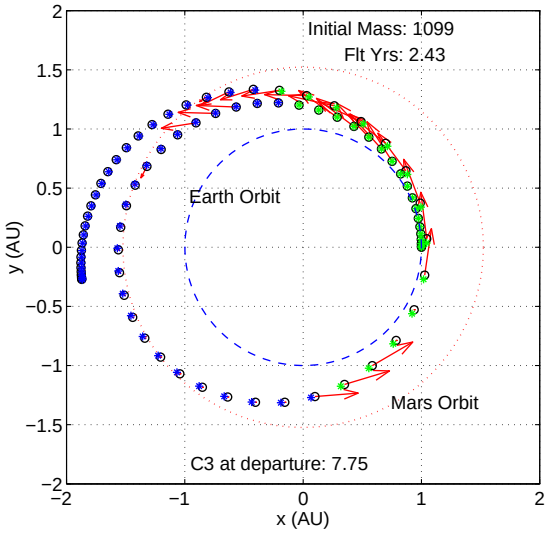

Fig. 3 Optimal orbit transfer for a SEP mission study.

are given in Table 2. Thus, the control Hamiltoninan for this problem (assuming no running cost) is given by,

$$H(\boldsymbol{\lambda},\mathbf{x},\mathbf{u}) = \left\langle \boldsymbol{\lambda}_v, \begin{pmatrix}\sin\theta\\ \cos\theta\end{pmatrix}\right\rangle \frac{T(P_e)}{m} - \lambda_m \gamma(P_e) + H_0$$

where H_0 are terms in the Hamiltonian that are independent of the control variables. It is now apparent that Problem HMC for a SEP mission design,

$$(\text{HMC: SEP}) \begin{cases} \underset{\mathbf{u}=(P_e,\theta)}{\text{Minimize}} & H(\boldsymbol{\lambda},\mathbf{x},\mathbf{u}) \\ \text{Subject to} & \mathbf{u} \in \mathbb{P} \times S^1 \end{cases}$$

has no apparent closed form solution. As discussed in Sec. 4, this immediately eliminates the HJB framework as a problem-solving tool.

As indicated elsewhere, during advanced mission design analysis, it is necessary to explore various trade spaces. These trade spaces can essentially be described in terms of Pareto sets but for the purposes of brevity, consider the simpler problem of maximizing the endpoint cost function,

$$J[\mathbf{x}(\cdot),\mathbf{u}(\cdot),t_0,t_f] = \alpha a_f + \beta e_f \qquad \alpha + \beta = 1$$

A solution for this problem using the CMP framework is shown in Fig. 3 for the case of $\alpha = \beta = 0.5$. This solution was obtained in [9] using the software package, DIDO,[24] which is an implementation of the CMP for the Legendre PS method. A plot of the thrust program and the steering angle is shown in Fig. 4. Note that the thrust is not at the maximum value over several segments of the trajectory. This is simply a direct effect of (20) and (25). It is also important to note that the CMP framework automatically determines the number and locations of the switches (if any). The solid line in Fig. 4 indicates the thrust obtained by an application of the the necessary conditions for Problem $HMC : SEP$ with the covectors determined from the CMP. The excellent agreement between the two solutions shows that the necessary conditions for Problem

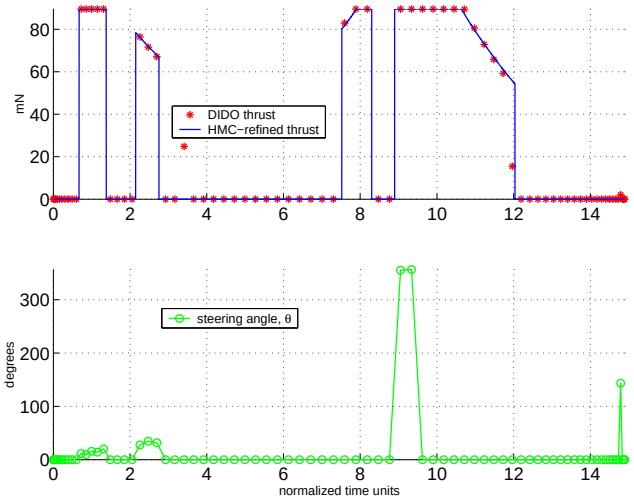

Fig. 4 Thrust vector program for a SEP mission.[9]

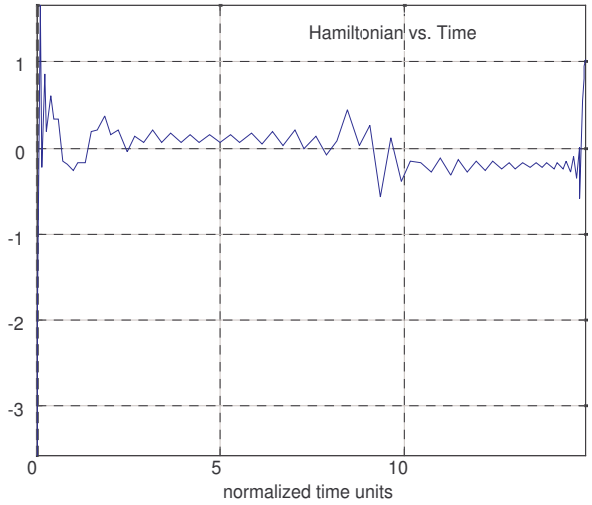

Fig. 5 Hamiltonian evolution for a SEP trajectory.

$HMC : SEP$ are being met at high precision. Many such optimality checks can be performed and some of these are described in [9]. One important check on the optimality of the trajectory is the Hamiltonian evolution equation, (19). By setting the time-dependent coefficients to zero, one can easily check the constancy of the Hamiltonian. As indicated in Sec. 7, this condition cannot be expected to be satisfied point wise if the totality of closure conditions are not included. That this is in fact true is demonstrated in Fig. 5. A cursory inspection of this figure shows that the Hamiltonian is weakly zero thereby satisfying an "inexact" condition on the vanishing of the Hamiltonian for time-free problems. Solutions such as these are typically computed in about 30 seconds (on desktop PCs) without even optimizing the code or the machine. By exploiting elementary features of the PS method, solutions to many problems can be obtained within fractions of a second.[25,27,31,34] Thus, it is obvious that optimal feedback orbit guidance by way of real-time optimal control is a clear modern-day reality.

Acknowledgments

I would like to acknowledge the generous funding for this research provided by NASA-GRC and the Secretary of the Air Force. In particular, I would like to thank John P. Riehl (NASA) and Stephen W. Paris (Boeing) for their enthusiastic support of pseudospectral methods for solving mission planning problems for NASA. My special gratitude goes to my former graduate students, Scott B. Josselyn and Patrick A. Croley, who showed me simple ways to use DIDO for poorly scaled problems.

References

[1]V. Adamchik and S. Wagon, "A Simple Formula for π," *American Mathematical Monthly*, Vol. 104, 1997, pp. 852-855.

[2]R. Andraka, "A Survey of CORDIC Algorithms for FPGAs," *Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays*, ACM, Inc., Feb. 22-24, 1998, Monterey, CA, pp. 191-200.

[3]U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, PA, 1998.

[4]M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhäuser, Boston, MA, 1997.

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

[6]A. E. Bryson and Y.-C. Ho, *Applied Optimal Control,* Hemisphere, New York, 1975.

[7]F. H. Clarke, *Optimization and Nonsmooth Analysis*, Classics in Applied Mathematics, SIAM, Philadelphia, PA, 1990.

[8]F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, *Nonsmooth Analysis and Control Theory,* Springer-Verlag, New York, NY, 1998.

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

[10]A. L. Dontchev, W. W. Hager and V. M. Veliov, "Second-Order Runge-Kutta Approximations in Control Constrained Optimal Control Problems," *SIAM Journal of Numerical Analysis*, Vo. 38, No. 1, 2000, pp. 202-226.

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

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

[13]Q. Gong, I. M. Ross, W. Kang, and F. Fahroo, "Convergence of Pseudospectral Methods for Constrained Nonlinear Optimal Control Problems," *Proc. The IASTED International Conference on Intelligent Systems and Control*, Honolulu, HI, pp. 209–214, 2004.

[14]W. W. Hager, "Runge-Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik,* Vol. 87, 2000, pp. 247-282.

[15]H. Halkin, "A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations," *SIAM Journal of Control*, Vol. 4, No. 1, 1966, pp. 90-111.

[16]S. I. Infeld, S. B. Josselyn, W. Murray and I. M. Ross, I. M., "Design and Control of Libration Point Spacecraft Formations," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Providence, RI, August 2004. AIAA Paper No. 2004-4786.

[17]S. Josselyn and I. M. Ross, "A Rapid Verification Method for the Trajectory Optimization of Reentry Vehicles," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 3, 2003.

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

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

[20]B. S. Mordukhovich, *Variational Analysis and Generalized Differentiation, I: Basic Theory*, vol. 330 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences] Series, Springer, Berlin, 2005.

[21]P. J. Nahin, *When Least is Best*, Princeton University Press, Princeton, NJ, 2004.

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

[23]L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko, *The Mathematical Theory of Optimal Processes*, Wiley-Interscience, New York, N.Y., 1962.

[24]I. M. Ross, *User's Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems*, Technical Report 04-01.0, Tomlab Optimization Inc, February 2004.

[25]I. M. Ross, "How to Find Minimum-Fuel Controllers," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Providence, RI, August 2004. AIAA Paper No. 2004-5346.

[26]I. M. Ross, "A Historical Introduction to the Covector Mapping Principle," *AAS/AIAA Astrodynamics Specialist Conference*, Tahoe, NV, August 8-11, 2005, Paper AAS 05-332.

[27]I. M. Ross and F. Fahroo, "Issues in the Real-Time Computation of Optimal Control," *Mathematical and Computer Modelling*, Vol. 40, Pergamon Publication (to appear).

[28]I. M. Ross and F. Fahroo, "A Perspective on Methods for Trajectory Optimization," *Proceedings of the AIAA/AAS Astrodynamics Conference*, Monterey, CA, August 2002. AIAA Paper No. 2002-4727.

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

[30]I. M. Ross, F. Fahroo and Q. Gong, "A Spectral Algorithm for Pseudospectral Methods in Optimal Control," *Proceedings of the 10th International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA)*, July 21-25, 2004, Orlando, FL, pp. 104-109.

[31]P. Sekhavat, A. Fleming and I. M. Ross, "Time-Optimal Nonlinear Feedback Control for the NPSAT1 Spacecraft," *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, AIM 2005, 2428 July 2005 Monterey, CA.

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

[33]R. B. Vinter, *Optimal Control*, Birkhäuser, Boston, MA, 2000.

[34]H. Yan, D. J. Lee, I. M. Ross, and K. T. Alfriend, "Real-Time Outer and Inner Loop Optimal Control Using DIDO," *AAS/AIAA Astrodynamics Specialist Conference*, Tahoe, NV, August 8-11, 2005, Paper AAS 05 - 353.