2002-07-09

Exploiting Higher-order Derivatives in Computational Optimal Control

Ross, I. Michael; Rea, Jeremy; Fahroo, Fariba

American Control Conference

https://hdl.handle.net/10945/29655

Proceedings of the 10th Mediterranean conference
on control and Automation - MED2002
Lisbon, Portugal, July 9-12, 2002.

EXPLOITING HIGHER-ORDER DERIVATIVES IN COMPUTATIONAL OPTIMAL CONTROL

I. Michael Ross[*], **Jeremy Rea**[†], **Fariba Fahroo**[‡]

[*] Department of Aeronautics and Astronautics, Mail Code: AA/Ro
Naval Postgraduate School
Monterey, CA 93943
e-mail:imross@nps.navy.mil

[†]Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139
email:jrea@mit.edu

[‡]Department of Mathematics, Code MA/Ff
Naval Postgraduate School
Monterey, CA 93943
e-mail:ffahroo@nps.navy.mil

Keywords: Nonlinear systems, optimal control, discretization, higher-order derivatives.

Abstract

To facilitate generation of real-time solutions to nonlinear optimal control problems, we present a new way of approximating higher-order derivatives that arise in control systems. A Legendre pseudospectral method is presented to efficiently and accurately discretize optimal control problems governed by higher-order dynamical constraints. For mechanical systems, a reduction in the number of unknown variables is immediately realized as a consequence of Newton's second law of motion which is of second order. The reduction in the size of the problem facilitates rapid solutions from nonlinear programming solvers. A rocket launch problem illustrates the differences in using standard state space first-order forms and second-order forms. The numerical results show that the second-order form generates faster results with increasing relative computational speed for increasing grid points.

1 Introduction

Over the last decade, numerical optimal control has developed to a mature field where the introduction of different discretization methods has made solving "hard" problems possible [1]. A current active area of research is to generate open-loop solutions in real-time so that nonlinear model predictive control is possible [13]. Most of these numerical methods belong to the class of *direct* methods which discretize and parameterize the optimal control problem into a Nonlinear Programming (NLP) problem. To get fast convergent solutions from the NLP solvers that use sequential quadratic programming algorithms, it has been desirable to reduce the size of the NLP variable. To this end, different reformulations of the original problem which result in elimination of some of the variables have been proposed. In the *inverse dynamic optimization method*, [2, 12], for example, the idea is to eliminate the control variable (when possible) by writing it in terms of the states and state rates. Also, in some recent work [8], the control and some components of the state variable are parameterized in terms of a given output variable and its derivatives. When the entire state variable (along with the control variable) can

be parameterized with respect to the output variable and its derivatives, then the system is called *differentially flat.* These systems have been extensively studied in [5]. One major disadvantage of the abovementioned methods is that they are only applicable to a rather narrow class of dynamical systems and optimal control problems. In any case, the transformations inherent in differentially flat systems require dealing with higher order derivatives of states, controls and the (flat) outputs. In many dynamical systems, higher-order derivatives occur quite naturally. For example, in mechanical systems, the dynamics are easily expressed in second-order form as a result of applying Newton's second law of motion. In [14], a framework has been presented to either preserve the higher-order form of the dynamics, or reduce the dimension of the state variable by eliminating some of its components by using fewer equations but of higher order forms. In all these analyses, the computational work has been based on approximating the variables by the polynomials (or splines) of appropriate orders. No framework for the discretization of the higher order dynamic equations have been presented.

In this paper, we use pseudospectral methods to handle the higher order derivatives. Pseudospectral methods have been proposed as an efficient way to solve a broad class of optimal control problems [3, 4, 10]. In particular, the Legendre pseudospectral method offers a very elegant way to check the optimality of the solution by way of the Covector Mapping Theorem [9]. As shown in Refs. [3] and [4], pseudospectral methods also offer a very simple way to solve problems governed by controlled differential inclusions including differential-algebraic equations.

Pseudospectral methods are distinguished from the "traditional" collocation schemes [1], in the way the dynamic constraints and the cost integral are discretized. The cost integral is typically discretized by Gauss quadratures whereas the derivative is approximated by way of a differentiation matrix. Pros and cons on using traditional methods versus pseudospectral methods are further discussed in Ref. [11]. The main purpose of this paper is to demonstrate how these methods can easily handle higher order differential equations. This in turn can result in a reduction of the size of the NLP which can yield faster solutions. A numerical example is included to illustrate the ideas.

2 Fundamental Principles

Consider a scalar-valued second-order controlled differential equation,

$$\ddot{y} = f(\dot{y}, y, u, t) \quad t \in [-1, 1] \tag{1}$$

where y and u are the scalar state and control variables. Traditionally, this equation is converted to "standard" phase-space coordinates

$$\dot{y}_1 = y_2 \tag{2}$$
$$\dot{y}_2 = f(y_2, y_1, u, t) \tag{3}$$

so that typical "first-order collocation" methods like Hermite-Simpson and Runge-Kutta can be applied. From a numerical point of view, one issue with increasing the dimension of the problem by the phase-space transformation is that it increases the number of unknown variables. Suppose that the time interval is divided into N segments so that the number of node points is $N + 1$. In the original second-order differential equation Eq. (1), the number of unknowns (from a collocation perspective) are $2(N+1)$ whereas in phase-space coordinates, the number of unknowns are $3(N + 1)$. From this simple example, it is evident that a direct discretization of the second-order equation reduces the number of unknown parameters by $N + 1$.

In pseudospectral methods, for any generic function $x(t)$ we define $X = [x_0, x_1, \ldots, x_N]^T$ which is the vector of values of $x(t)$ at the node points. We use a differentiation matrix $\mathbf{D}$ that operates on X to generate approximations to the derivatives, $\dot{x}_0, \dot{x}_1, \ldots, \dot{x}_N$ at the node points, $T = [t_0, t_1, \ldots, t_N]$. In the Legendre Pseudospectral Method [9], which is the method used here, the node points are Legendre-Gauss-Lobatto (LGL) points. These points are used extensively in quadrature formulas and their nonuniform distribution (with highest density towards the end-points) give the least interpolation error in the L^2-norm. These points $t_l, l = 0, \ldots, N$ are given by

$$t_0 = -1, \quad t_N = 1$$

and for $1 \leq l \leq N-1$, t_l are the zeros of $\dot{L}_N$, the derivative of the Legendre polynomial, L_N.

For the Legendre Pseudospectral Differentiation Matrix, it can be shown [7] that the square of the differentiation matrix, $\mathbf{D}^2$, operates on X and provides excellent approximations to the second derivatives at the LGL points. Hence, Eq.(1) can be approximated as,

$$\mathbf{D}^2 Y = F(\mathbf{D}Y, Y, U, T) \tag{4}$$

where $F = [f_0, f_1, \ldots, f_N]^T$ is the discretization (vector) of f.

3 Description of the Method

Problem Formulation

Consider the following optimal control problem. Determine the trajectory-control pair, $[\tau_0, \tau_f] \ni \tau \mapsto \{\mathbf{x} \in R^n, \mathbf{u} \in R^m\}$ that minimizes the Mayer cost functional,

$$J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), \tau_f] = E\left(\mathbf{x}(\tau_f), \tau_f\right) \tag{5}$$

where $E : R^n \times R \to R$. For notational simplicity we assume the dynamical constraints to be given by the following second order system

$$\begin{aligned} \ddot{\mathbf{x}}(\tau) &= \mathbf{f}(\dot{\mathbf{x}}(\tau), \mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \\ \mathbf{f} &: R^n \times R^n \times R^m \times R \to R^n \end{aligned} \tag{6}$$

It will be apparent that the extension to higher-order systems is trivial. In addition, we assume the boundary conditions and the path constraints to be given by,

$$\mathbf{e}_l \leq \mathbf{e}(\mathbf{x}(\tau_0), \tau_0, \mathbf{x}(\tau_f), \tau_f) \leq \mathbf{e}_u \tag{7}$$

$$\mathbf{g}_l \leq \mathbf{g}\left(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau\right) \leq \mathbf{g}_u \tag{8}$$

where $\mathbf{e} : R^n \times R \times R^n \times R \to R^p$ and $\mathbf{g} : R^n \times R^m \times R \to R^r$. The constant vectors $\mathbf{e}_l, \mathbf{e}_u \in R^p$ represent the lower and upper bounds of these inequalities. Similarly, $\mathbf{g}_l, \mathbf{g}_u \in R^r$ denote the lower and upper bounds of $\mathbf{g}$. Clearly, an equality constraint may be obtained by simply setting the lower and upper bounds to be equal.

Since the node points (LGL) lie in the computational interval $[-1, 1]$, the problem is transformed to this interval by the linear transformation for $t \in [t_0, t_N] = [-1, 1]$

$$\tau = \frac{(\tau_f - \tau_0)t + (\tau_f + \tau_0)}{2} \tag{9}$$

resulting in the following reformulation of Eqs. (5), (6), (7), (8)

$$J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), \tau_f] = E\left(\mathbf{x}(1), \tau_f\right) \tag{10}$$

$$\left(\frac{2}{\tau_f - \tau_0}\right)^2 \ddot{\mathbf{x}}(t) = \mathbf{f}\left[\left(\frac{2}{\tau_f - \tau_0}\right)\dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{u}(t), \tau(t)\right] \tag{11}$$

$$\mathbf{e}_l \leq \mathbf{e}(\mathbf{x}(-1), \tau_0, \mathbf{x}(1), \tau_f) \leq \mathbf{e}_u \tag{12}$$

$$\mathbf{g}_l \leq \mathbf{g}(\mathbf{u}(t), \mathbf{x}(t), \tau(t)) \leq \mathbf{g}_u \tag{13}$$

Direct Approximation Method

In the Legendre pseudospectral method, we seek polynomial approximations for the state and control functions at the LGL points,

$$\mathbf{x}^N(t) = \sum_{l=0}^{N} \mathbf{x}_l \phi_l(t), \quad \mathbf{u}^N(t) = \sum_{l=0}^{N} \mathbf{u}_l \phi_l(t) \tag{14}$$

where, $\mathbf{x}_l = \mathbf{x}^N(t_l)$; $\mathbf{u}_l = \mathbf{u}^N(t_l)$ and for $l = 0, 1, ..., N$, $\phi_l(t)$ are the Lagrange polynomials of order N :

$$\phi_l(t) = \frac{1}{N(N+1)L_N(t_l)} \frac{(t^2-1)\dot{L}_N(t)}{t - t_l} \tag{15}$$

The derivatives $\dot{\mathbf{x}}(t)$ and $\ddot{\mathbf{x}}(t)$ are approximated by $\dot{\mathbf{x}}^N(t)$ and $\ddot{\mathbf{x}}^N(t)$, respectively. Evaluating them at the LGL node points t_k results in matrix multiplications of the following form [7]

$$\dot{\mathbf{x}}^N(t_k) = \sum_{l=0}^{N} D_{1,kl} \mathbf{x}_l \tag{16}$$

$$\ddot{\mathbf{x}}^N(t_k) = \sum_{l=0}^{N} D_{2,kl} \mathbf{x}_l \tag{17}$$

where $D_{1,kl} = \frac{d\phi_l}{dt}(t_k)$ are entries of the $(N+1) \times (N+1)$ first-order differentiation matrix $\mathbf{D}^1$:

$$\mathbf{D^1} := [D_{1,kl}] := \begin{cases} \frac{L_N(t_k)}{L_N(t_l)} \cdot \frac{1}{t_k - t_l} & k \neq l \\ -\frac{N(N+1)}{4} & k = l = 0 \\ \frac{N(N+1)}{4} & k = l = N \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

The elements of the second-order differentiation matrix $D_{2,kl} = \frac{d^2\phi_l}{dt^2}(t_k)$ can be easily computed by squaring $\mathbf{D}^1$, i.e., $\mathbf{D}^2 = \mathbf{D}^1 * \mathbf{D}^1$.

From approximations of the Eqs (10-13), the following NLP is obtained: Find coefficients

$$\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N),\ \mathbf{U} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_N)$$

and possibly the final time τ_f to minimize

$$J^N(\mathbf{X}, \tau_f) = E(\mathbf{x}_N, \tau_f) \tag{19}$$

subject to

$$\left(\frac{2}{\tau_f - \tau_0}\right)^2 \sum_{l=0}^{N} D_{2,kl}\mathbf{x}_l$$
$$-\mathbf{f}\left(\left(\frac{2}{\tau_f - \tau_0}\right)\sum_{l=0}^{N} D_{1,kl}\mathbf{x}_l, \mathbf{u}_k, \tau(t_k)\right) = \mathbf{0} \tag{20}$$

$$\mathbf{e}_l \leq \mathbf{e}(\mathbf{x}_0, \tau_0, \mathbf{x}_N, \tau_f) \leq \mathbf{e}_u \tag{21}$$
$$\mathbf{g}_l \leq \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k, \tau(t_k)) \leq \mathbf{g}_u \tag{22}$$

for $k = 0, \ldots, N$.

4 A Numerical Example

Consider a two-dimensional launch problem. It is desired to reach a certain altitude and final velocity condition while maximizing the final mass; thus, the cost function is,

$$J = -m(\tau_f) \tag{23}$$

From Newton's Laws of motion and gravity, the exo-atmospheric dynamical equations that define the system are,

D^2 **Form**

$$\ddot{x} = \frac{T_x}{m} \tag{24}$$
$$\ddot{y} = \frac{T_y}{m} - \frac{g_o}{\left(1 + \frac{y}{R}\right)^2} \tag{25}$$
$$\dot{m} = -\frac{T}{g_o I_{sp}} \tag{26}$$

where x, y and m are downrange, altitude and mass, respectively, and T, the thrust magnitude, is constrained by

$$0 \leq T = \sqrt{T_x^2 + T_y^2} \leq T_{max} \tag{27}$$

where T_x and T_y are the components of the thrust vector along the x and y directions, respectively. The other parameters g_0, R and I_{sp} are constants which are the gravitational acceleration at the surface of Earth, the radius of Earth, and the specific impulse of the rocket, respectively.

The boundary conditions are given by

$$x(0) = \dot{x}(0) = y(0) = \dot{y}(0) = 0$$
$$x(\tau_f) = \text{ free},\ \tau_f = \text{ free}$$

The remainder of the boundary conditions along with the constant parameters are given in Table 1 below.

Final Altitude (m)	10000
Final X-velocity (m/s)	1000
Final Y-velocity (m/s)	150
Initial Mass (kg)	5000
Propellant Mass Fraction	0.6
Maximum Thrust to Initial Weight Ratio	2
Specific Impulse (s)	300
g_0 (m/s^2)	9.8203

Table 1: Parameters for the Numerical Example

This problem can be discretized as formulated by using the results developed in this paper. In the "state-space form" referred to as the D form here, the dynamic constraints must be reformulated as

D **Form**

$$\dot{x} = V_x \tag{28}$$
$$\dot{y} = V_y \tag{29}$$
$$\dot{V}_x = \frac{T_x}{m} \tag{30}$$
$$\dot{V}_y = \frac{T_y}{m} - \frac{g_0}{\left(1+\frac{y}{R}\right)^2} \tag{31}$$
$$\dot{m} = -\frac{T}{g_0 I_{sp}} \tag{32}$$

The following scaling parameters are used to numerically balance the problem; all other units are defined with these units.

- Unit of Length: Radius of Earth = 6371 km
- Unit of Velocity: Circular orbital velocity at surface of the Earth = 7.9098 km/s
- Unit of Mass: Initial mass of vehicle = 5000 kg

Both forms of the problem are discretized and solved using the Legendre Pseudospectral Method described above. The optimization vectors for each form are given below. These vectors are made up of the values of the approximate states and controls at the LGL nodes, and the free final time.

D **Form**

$$\vec{x}_{opt} = \left[\begin{array}{cccccccc} \vec{x}^T & \vec{y}^T & \vec{V}_x^T & \vec{V}_y^T & \vec{m}^T & \vec{T}_x^T & \vec{T}_y^T & \tau_f \end{array} \right]^T \tag{33}$$

The size of the optimization vector is $7n_{LGL} + 1$, where n_{LGL} is the number of Legendre-Gauss-Lobatto points used to discretize the time domain of the problem.

D^2 **Form**

$$\vec{x}_{opt} = \left[\begin{array}{cccccc} \vec{x}^T & \vec{y}^T & \vec{m}^T & \vec{T}_x^T & \vec{T}_y^T & \tau_f \end{array} \right]^T \tag{34}$$

The size of the optimization vector is $5n_{LGL} + 1$.

Several cases were run using each form described above. The problem was set-up using MATLAB, version 5.3.1. The nonlinear program solver NPSOL [6] was used as the numerical optimizer. The problem was run on a Sun Microsystems Enterprise 450 Model 4400, with 4 Sun UltraSPARC-II 400 MHz CPU's running under SunOS 5.6 as an operating system. In each case, the optimizer was limited to a maximum of 100 iterations.

Figure 1 shows the difference in the size of the problem for the D and D^2 forms as a function of the number of LGL discretization points. Figure 2 shows the solution times for each method as a function of LGL discretization points. Note that the D^2 form is faster than the D form. When a small number of LGL points are used to discretize the problem, the difference in the problem size and convergence times between the D and D^2 forms are not great. However, as more points are added, the difference in the problem size increase, and the difference in convergence times increases. So, while the D^2 form has no great advantage over the D form for small numbers of LGL points, it does have an increasing advantage for larger numbers of LGL points.

Figures 3 through 6 show a sample solution using 50 LGL points. Note how closely the D- and D^2-solutions match indicating that there are no significant differences in the final solution. In other words, both methods give the same numerical results but the D^2 Form runs faster.

5 Conclusions

It is apparent that the Legendre Pseudospectral Method described above can be used quite effectively to reduce the size of the NLPs resulting from discretized optimal control problems. This size reduction increases the computational speed of convergence. The relative performance of the size reduction increases with increasing number of discretization points. This reduction also allows the dynamical equations to be written in their natural forms. For mechanical systems, the size reduction is offered directly from Newton's Laws of motion, involving second-order differential equations for position. An extension to higher order derivatives is straight-forward.

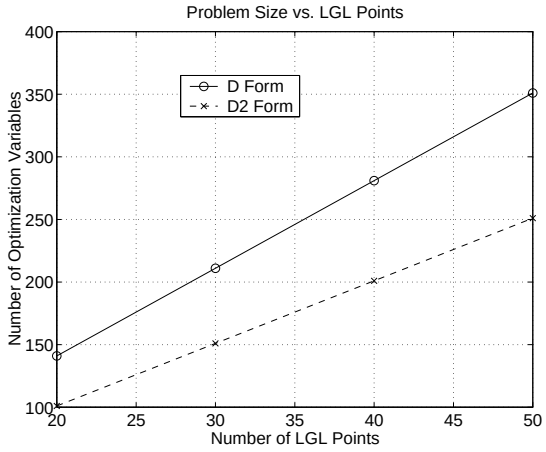

Figure 1: Problem Size

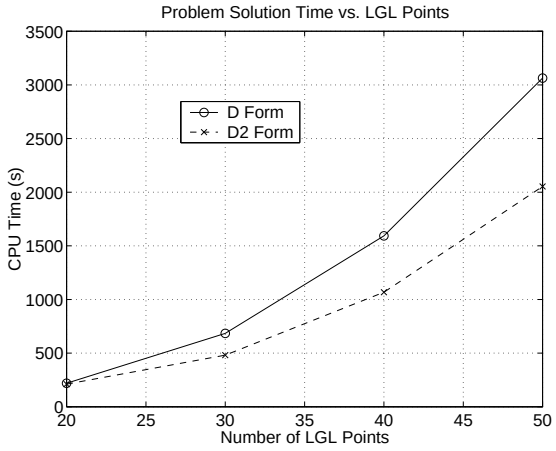

Figure 2: Convergence Time Comparison

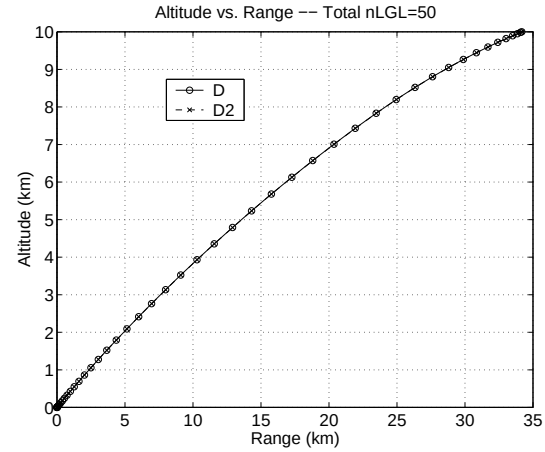

Figure 3: Vehicle Trajectory

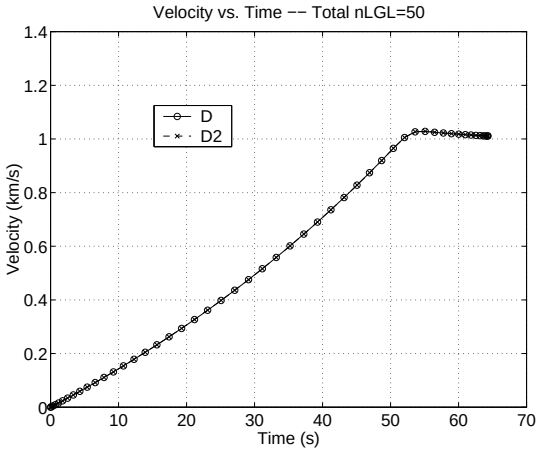

Figure 4: Time History of Velocity

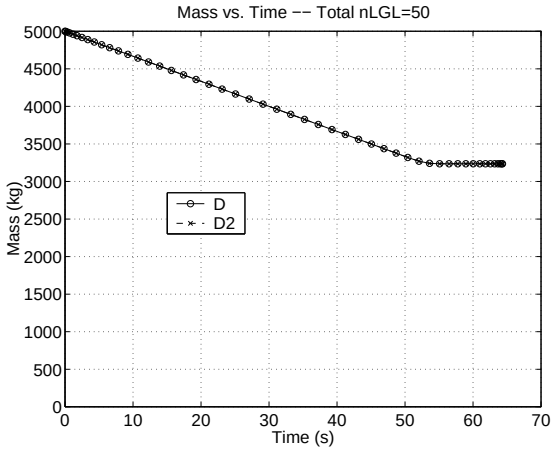

Figure 5: Time History of Mass

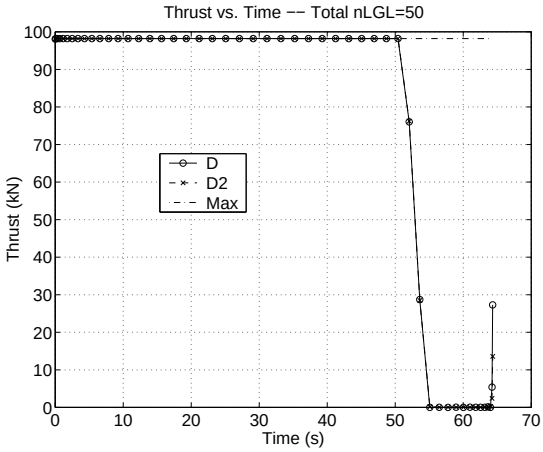

Figure 6: Time History of Thrust

Acknowledgments

We gratefully acknowledge funding for this research provided in part by The Charles Stark Draper Laboratory and the Naval Postgraduate School.

References

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

[2] A. E. Bryson. *Dynamic Optimization,* Addison-Wesley Longman Inc., Menlo Park, CA, (1999).

[3] F. Fahroo and I. M. Ross. "Second look at approximating differential inclusions," *Journal of Guidance, Control and Dynamics,* Vol. 24, No. 1, pp. 131-133, (2001).

[4] F. Fahroo and I. M. Ross. "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control and Dynamics,* Vol. 25, No. 1, pp. 160-166, (2002).

[5] M. Fliess, J. Levine, P. Martin and P. Rouchon. "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control,* 61(6), June, pp. 1327-1361, (1995).

[6] P. E. Gill, W. Murray, M. A. Saunders and M. A. Wright. "User's guide for NPSOL 5.0: A Fortran package for nonlinear programming," Technical Report SOL 86-1, July, (1998).

[7] D. Gottlieb, M. Y. Hussaini and S. A. Orszag. "Theory and applications of spectral methods," *Spectral Methods for PDE's,* SIAM, Philadelphia, PA, (1984).

[8] N. Petit, M. B. Milam and R. M. Murray. " Inversion based constrained trajectory optimization," *Proceedings of the 2001 IFAC Symposium on Nonlinear Control Systems Design,* (2001).

[9] I. M. Ross and F. Fahroo. "A pseudospectral transformation of the covectors of optimal control systems," *Proceedings of the First IFAC Symposium on Systems, Structure and Control,* Prague, the Czech Republic, August 2001.

[10] I. M. Ross and F. Fahroo. "A direct method for solving nonsmooth optimal control problems," to appear in the *Proceedings of the 2002 IFAC World Congress* , Barcelona, Spain, July 2002.

[11] I. M. Ross and F. Fahroo. "A perspective on methods for trajectory optimization," Invited Paper; to appear in the *Proceedings of the AIAA/AAS Astrodynamics Conference,* Monterey, CA, August 2002.

[12] H. Seywald. "Trajectory optimization based on differential inclusion," *Journal of Guidance, Control and Dynamics,* Vol. 17, No. 3, pp. 480-487, (1994).

[13] J. Strizzi, I. M. Ross, and F. Fahroo. "Towards real-time computation of optimal controls for nonlinear systems," to appear in *Proceedings of the AIAA Guidance Navigation and Control Conference,* Monterey, CA, August 2002.

[14] T. Veeraklaew and S. K. Agrawal. "New computational framework for trajectory optimization of higher-order dynamic systems," *Journal of Guidance, Control and Dynamics,* Vol. 24, No. 2, pp. 228-236, (2001).