

Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems

I. Michael Ross¹
Fariba Fahroo²

Abstract

This paper presents some preliminary results on combining two new ideas from nonlinear control theory and dynamic optimization. We show that the computational framework facilitated by pseudospectral (PS) methods applies quite naturally and easily to the Lie-Bäcklund equivalence of nonlinear controlled dynamical systems. The optimal motion planning problem for differentially flat systems is equivalent to a classic Bolza problem of the calculus-of-variations. In this paper, we exploit the notion that derivatives of flat outputs given in terms of Lagrange interpolating functions can be quickly and easily computed using PS differentiation matrices. The application of this method to the crane control problem demonstrates how flatness may be readily exploited. In the case of partial dynamic inversion, or systems with non-zero defects, differential constraints are satisfied at optimal node points whose optimality criterion is some error norm. Integral cost functionals are handled by Gauss-type quadrature rules. Numerical experiments suggest that PS methods are superior to other methods that exploit full or partial dynamic inversion; however, a number of problems inherent to utilizing flat outputs to real-time trajectory optimization remain open.

Keywords: Differential flatness, optimal control theory, pseudospectral methods

1 Introduction

A significant number of future military systems are based on autonomous operation of multiple agents working in a cooperative and non-cooperative manner to achieve optimal mission performance. The design of these proposed new systems demand peak performance and autonomy over a wide range of operating conditions. Examples of these range from the formation-keeping and reconfiguration of satellite swarms to abort

guidance of a reusable launch vehicle to the design of minimum-time maneuvers for unmanned combat air vehicles. In all of these problems, the systems are nonlinear in both the state and control variables. Linearizing or simplifying the system dynamics often leads to substandard performance. The proper framework for addressing many of these problems is nonlinear optimal control theory. Despite significant progress in this field over the last 40 years, “... solving optimal control problems of scientific or engineering interest is often extremely difficult.” [17] In recent years, a new family of pseudospectral (PS) methods have emerged as a viable option for real-time trajectory optimization [3, 4, 5, 13]. PS methods are quite unique in the sense that they treat the approximation of the tangent bundle separately from the approximation of the vector field. Thus, they can readily exploit new ideas and concepts that rely on differential-geometric properties of control systems. In this paper, we explore harnessing the concept of differential flatness [7, 8] towards solving the problem of real-time trajectory generation for nonlinear systems.

Differential flatness of nonlinear control systems was introduced by Fliess et al. [7] as part of a notion that certain differential algebraic representation of dynamical systems are equivalent [8]. Roughly speaking, a dynamical system is differentially flat if it is equivalent to a system without dynamics, i.e. a static system. Although it is difficult to determine whether a system is differentially flat, a growing number of problems in engineering have been shown to be flat, (see Ref. [8] and the references contained therein). For a flat system, the motion planning problem simply reduces to finding a sufficiently smooth output, $t \mapsto \mathbf{y}(t)$, that satisfies the boundary conditions in output space. In principle, finding such smooth functions is not difficult, since the output can be represented in terms of polynomials with unknown coefficients. These coefficients can then be determined by imposing the condition that the polynomials and their derivatives should automatically satisfy the boundary conditions in output space [6, 7, 8]. However, when differentiating polynomials, it is extremely important to be cognizant of instabilities like the Runge phenomenon [16]. Further, as noted earlier, in many high-performance systems, it is not enough to find feasible trajectories but trajectories

¹Associate Professor, Department of Aeronautics and Astronautics, Code AA/Ro, Naval Postgraduate School, Monterey, CA 93943, E-mail:imross@nps.navy.mil.

²Associate Professor, Department of Mathematics, Code Ma/Ff, Naval Postgraduate School, Monterey, CA 93943, E-mail:ffahroo@nps.navy.mil.

that optimize a Bolza cost functional. For differentially flat systems, the optimal control problem is equivalent to a classic unconstrained calculus-of-variations problem [1]. Note that this equivalence facilitated by the Lie-Bäcklund transformation[8] is quite different from the neo-classical approach where the controls are eliminated by a passage through a differential inclusion to a generalized Bolza problem [2].

The states and controls of flat systems can be represented by $\mathbf{x} = \mathbf{a}(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(\beta)})$, $\mathbf{u} = \mathbf{b}(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(\beta+1)})$, where $\mathbf{y}^{(r)}$ is the r th derivative of \mathbf{y} . PS methods are particularly suited for such representations since the derivatives of \mathbf{y} can be quickly and easily obtained by the repeated use of differentiation matrices. In the case of optimal motion planning, approximations may arise if the cost function is given in an integral form (since an integral cost is equivalent to adding dynamics). However, since PS methods are based on Gauss-type quadrature rules, exact performance representation is possible for certain families of integrands (e.g. polynomials). Much of our attention in this paper is based on the use of orthogonal polynomials and their applications towards fully invertible (i.e. flat) systems. Our methods easily extend to partial inversion and flat systems with inequality constraints (see for example, Refs. [4, 15]).

Our work is similar in spirit to that of Milam et al. [11] and Petit et al. [12] but we will show that our technique is markedly different yet simpler to implement than their B-spline approach. Further, since differentially flat systems are sufficiently smooth, PS approximations for functions and their derivatives provide exponential convergence (accuracy) rates for analytic functions (outputs, states and inputs) and $O(N^{-m})$ for every m for C^∞ functions [16]. This property (known as spectral accuracy) is particularly important for systems where flat outputs cannot be obtained but an output that partially inverts the dynamics can be found and exploited. In this case, the dynamical constraints can be reduced but not eliminated while maintaining spectral accuracy [14]. Potential convergence problems resulting from discretizing the transformed dynamics are handled well by PS methods.

2 Problem Formulations

A classic smooth optimal control problem can be stated as:

Problem C

Determine the trajectory-control pair, $[\tau_0, \tau_f] \ni \tau \mapsto \{\mathbf{x} \in \mathbb{R}^{N_x}, \mathbf{u} \in \mathbb{R}^{N_u}\}$ and possibly the clock times τ_0

and τ_f , that minimize the Bolza cost functional,

$$J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), \tau_0, \tau_f] = E(\mathbf{x}(\tau_0), \mathbf{x}(\tau_f), \tau_0, \tau_f) + \int_{\tau_0}^{\tau_f} F(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \quad (1)$$

subject to the classic dynamic constraints,

$$\dot{\mathbf{x}}(\tau) = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) \quad \forall \tau \in (\tau_0, \tau_f) \quad (2)$$

and end point constraints,

$$\mathbf{e}_l \leq \mathbf{e}(\mathbf{x}(\tau_0), \mathbf{x}(\tau_f), \tau_0, \tau_f) \leq \mathbf{e}_u \quad (3)$$

For simplicity in presentation, we assume all functions to be C^∞ -smooth. The dynamical system described by Eq.(2) is differentially flat[7] if there exists a variable $\mathbf{y} \in \mathbb{R}^{N_u}$ and a function $\mathbf{c}(\cdot)$,

$$\mathbf{y} = \mathbf{c}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^{(\alpha)}) \quad (4)$$

such that

$$\mathbf{x} = \mathbf{a}(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(\beta)}) \quad (5)$$

$$\mathbf{u} = \mathbf{b}(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(\beta+1)}) \quad (6)$$

where α and β are finite positive integers that denote the number of derivatives of the respective variables. The variable, \mathbf{y} is called a flat or linearizing output. For ease of notation, we let $s = \beta + 1$ and denote the flat output and its derivatives by the composite variable, $\mathbf{z} \in \mathbb{R}^{(s+1)N_u}$,

$$\mathbf{z} = [\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(s)}]^T \quad (7)$$

so that $\mathbf{a}(\cdot) : \mathbf{z} \rightarrow \mathbf{x}$ and $\mathbf{b}(\cdot) : \mathbf{z} \rightarrow \mathbf{u}$. For a differentially flat system, Problem C can now be replaced by,

Problem DF

Determine the smooth function, $[\tau_0, \tau_f] \ni \tau \mapsto \mathbf{y} \in \mathbb{R}^{N_u}$, and possibly the clock times, τ_0 and τ_f , that minimize the classic Bolza cost functional [1],

$$\tilde{J}[\mathbf{y}(\cdot), \tau_0, \tau_f] = \tilde{E}(\mathbf{z}(\tau_0), \mathbf{z}(\tau_f), \tau_0, \tau_f) + \int_{\tau_0}^{\tau_f} \tilde{F}(\mathbf{z}(\tau)) d\tau \quad (8)$$

subject to the end point constraints,

$$\tilde{\mathbf{e}}_l \leq \tilde{\mathbf{e}}(\mathbf{z}(\tau_0), \mathbf{z}(\tau_f), \tau_0, \tau_f) \leq \tilde{\mathbf{e}}_u \quad (9)$$

where $\tilde{E}(\cdot)$, $\tilde{F}(\cdot)$ and $\tilde{\mathbf{e}}(\cdot)$ denote functions obtained from $E(\cdot)$, $F(\cdot)$ and $\mathbf{e}(\cdot)$ respectively by an appropriate substitution of Eqs.(5) and (6) in Eqs.(1) and (3). Of course, by the definition of differential flatness, Eq.(2) is automatically satisfied and hence is not a constraint. Thus, Problems C and DF are equivalent although their trajectories live on different manifolds with different dimensions.

Now suppose that \mathbf{y} is not a linearizing output (for example, if the system is not flat), then Problem DF includes additional dynamic constraints of the form,

$$\tilde{\mathbf{f}}(\mathbf{z}(\tau)) = \mathbf{0} \quad (10)$$

PS methods for solving such problems are discussed in [15]; in this paper, we focus on using the Legendre PS method that explicitly utilizes flat outputs.

3 The Legendre Pseudospectral (PS) Method

For the purpose of clarity and brevity, we discuss only the Legendre PS method. The basic underlying principles for other PS methods (e.g. Chebyshev [5]) are same; however, their numerical properties are different.

Let $L_N(t)$ be the Legendre polynomial of degree N . The Legendre-Gauss-Lobatto (LGL) points, $t_l, l = 0, \dots, N$ are given by $t_0 = -1, t_N = 1$, and for $1 \leq l \leq N-1$, t_l are the zeros of L'_N , the derivative of $L_N(t)$. The choice of these points provides us with optimal results (in the L^2 -sense) for interpolation of functions and prevents the Runge phenomenon [16]. For Problem C, the Legendre PS method offers an approximation for evaluating the integral by Gauss quadratures while the differential constraint is approximated by driving the residuals to zero at the LGL points. In this manner, the Legendre PS method unifies the discretization of both the integrals and the derivatives, and in both cases the discretizations are highly accurate. Further details on the approximation method for Problem C is described in [3, 5, 13]. Here, we focus our attention to Problem DF and the transformations necessary to cast Problem C to this format.

Since the LGL node points lie in the computational interval $[-1, 1]$, the linear transformation, $2\tau = (\tau_f - \tau_0)t + (\tau_f + \tau_0)$ is used to scale the domain, $[\tau_0, \tau_f]$. Next, the vector-valued function, $t \mapsto \mathbf{y}(t)$, is written as some N th degree vector-valued polynomial of the form

$$\mathbf{y}(t) = \sum_{l=0}^N \mathbf{y}_l \phi_l(t) \quad (11)$$

where, $\mathbf{y}_l := \mathbf{y}(t_l)$ are the unknown coefficients, and for $l = 0, 1, \dots, N$

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

are the Lagrange polynomials of order N that satisfy the Kronecker identity, $\phi_l(t_k) = \delta_{lk}$, where $\delta_{lk} = 1$ for $l = k$ and is zero otherwise. The composite variable \mathbf{z}

is then obtained simply by differentiating Eq.(11),

$$\dot{\mathbf{y}}(t) = \sum_{l=0}^N \mathbf{y}_l \dot{\phi}_l(t) \quad (12)$$

\vdots

$$\mathbf{y}^{(s)}(t) = \sum_{l=0}^N \mathbf{y}_l \phi_l^{(s)}(t) \quad (13)$$

where as before the superscript s denotes the s th derivative. It is apparent that we must choose $N \geq s + 1$. Evaluating the derivatives at t_k results in a matrix multiplication of the following form:

$$\dot{\mathbf{y}}(t_k) = \sum_{l=0}^N D_{1,kl} \mathbf{y}_l \quad (14)$$

\vdots

$$\mathbf{y}^{(s)}(t_k) = \sum_{l=0}^N D_{s,kl} \mathbf{y}_l \quad (15)$$

where $D_{i,kl}$, $i = 1, \dots, s$ are the entries of $(N+1) \times (N+1)$ differentiation matrices \mathbf{D}_i . The matrix, \mathbf{D}_1 is given by [4]

$$\mathbf{D}_1 := [D_{1,kl}] := \begin{cases} \frac{L'_N(t_k)}{L'_N(t_l)} \cdot \frac{1}{t_k - t_l} & k \neq l \\ -\frac{N(N+1)}{4} & k = l = 0 \\ \frac{N(N+1)}{4} & k = l = N \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

It can be shown that $\mathbf{D}_i = \mathbf{D}^i$ where the superscript denotes matrix powers. Thus, \mathbf{D}_2 is obtained by simply squaring \mathbf{D}_1 , while $\mathbf{D}_3 = \mathbf{D}^3$ and so on. Since $\mathbf{Y} = [\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N] \in \mathbb{R}^{N_u \times (N+1)}$ is an equivalent representation of the vector-valued polynomial given by Eq.(11), it follows that,

$$\mathbf{Y}_i = [\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N] [\mathbf{D}^i]^T \quad (17)$$

is an equivalent representation of the vector-valued polynomials, $\mathbf{y}^{(i)}$, $i = 1, \dots, s$ given by Eq.(12). In other words, the derivatives of the flat outputs at the LGL points are obtained by a simple matrix multiplication of the flat output with the appropriate order of the differentiation matrix. This is better illustrated as follows: Let $\mathbf{Z} = [\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_N] \in \mathbb{R}^{(s+1)N_u \times (N+1)}$ (see Eq.(7)). Then,

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Y}_0 \\ \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_s \end{bmatrix} = (\mathcal{E} \otimes [\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N]) \mathbf{D} \quad (18)$$

where \mathcal{E} is a $(s+1) \times 1$ vector of ones, \otimes denotes the Kronecker product and \mathbf{D} is a $(s+1)(N+1) \times (s+1)(N+1)$ block diagonal matrix where each block is $(N+1) \times (N+1)$ and the $(s+1)$ block diagonal entries are given by $[\mathbf{D}^i]^T$, $i = 0, 1, \dots, s$. An interesting situation arises when the clock times are fixed and the end point constraints in output space are given by linear inequalities of the form, (Cf. Eq.(9))

$$\tilde{\mathbf{e}}_l \leq \mathbf{A}[\mathbf{z}(\tau_0) \ \mathbf{z}(\tau_f)] \leq \tilde{\mathbf{e}}_u \quad (19)$$

where \mathbf{A} is a matrix of appropriate dimension. The motion planning problem is now reduced to solving a linear matrix inequality of finding the $N_u \times (N+1)$ parameters $[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N]$ such that

$$\tilde{\mathbf{e}}_l \leq \mathbf{B}[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N] \leq \tilde{\mathbf{e}}_u \quad (20)$$

where \mathbf{B} is obtained from Eqs.(18) and (19). Recall that N is a design parameter and must be chosen such that $N \geq s+1$. For a point-to-point motion planning problem in output space, Eq.(20) reduces to simply solving a full-rank linear matrix equation for $N = s+1$, which can obviously be done in real-time. However, a better alternative might be to choose $N \gg s$ and determine the extra degrees of freedom by minimizing some cost functional. In any case, the optimal motion planning problem requires that the integral given in Eq.(8) be evaluated in terms of the values of the flat outputs and its derivatives at the LGL points. While other polynomial approximations [11] can only use low-order quadrature schemes, in PS methods, high-order quadrature rules such as the Gauss-Lobatto integration rule can be naturally employed. The integral given by Eq.(8) is approximated by a finite sum which is exact for integrands which are polynomials of degree $2N-1$:

$$\tilde{J}[\mathbf{Y}, \tau_0, \tau_f] \approx \tilde{E}^N(\mathbf{z}_0, \mathbf{z}_N, \tau_0, \tau_f) + \sum_{k=0}^N \tilde{F}^N(\mathbf{z}_k) w_k \quad (21)$$

where w_k are the LGL weights,

$$w_k := \frac{2}{N(N+1)} \frac{1}{[L_N(t_k)]^2} \quad k = 0, 1, \dots, N \quad (22)$$

Thus, Problem DF is discretized to the following mathematical programming problem:

Problem DF^N

Find the $N_u \times (N+1)$ vector $\mathbf{Y} = [\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N]$ and possibly τ_0 and τ_f that minimize

$$\tilde{J}^N[\mathbf{Y}, \tau_0, \tau_f] = \sum_{k=0}^N \tilde{F}(\mathbf{z}_k) w_k + \tilde{E}(\mathbf{z}_0, \mathbf{z}_N, \tau_0, \tau_f) \quad (23)$$

subject to

$$\tilde{\mathbf{e}}_l \leq \tilde{\mathbf{e}}(\mathbf{z}_0, \mathbf{z}_N, \tau_0, \tau_f) \leq \tilde{\mathbf{e}}_u \quad (24)$$

If \tilde{F} and \tilde{E} are linear in \mathbf{z}_k , then the problem reduces to a linear programming problem for linear conditions in output space. In general, this is a nonlinear programming problem which can be solved using commercial off-the-shelf packages like SNOPT [9]. It is worth noting that in our method, the original state and control variables can be easily recovered by using the differentiation matrix and the functions $\mathbf{a}(\cdot)$ and $\mathbf{c}(\cdot)$.

4 Example: The Crane Problem

We first illustrate our approach by way of the crane example from [7]. A two-dimensional state model of a trolley-load of a crane is given by,

$$m\ddot{x} = -T \sin \theta \quad (25)$$

$$m\ddot{z} = -T \cos \theta + mg \quad (26)$$

$$\dot{x} = R \sin \theta + D \quad (27)$$

$$\dot{z} = R \cos \theta \quad (28)$$

where (x, z) are the coordinates of the load, m , which is connected to a trolley by a rope of length R and tension T . The trolley is at some distance D along the x -axis while the load is at an angle θ away from the vertical. As shown in Ref. [7], the system is differentially flat with a linearizing output given by $\mathbf{y} = [x, z]^T$.

The basic control problem is to carry the load m from (R_1, D_1) to (R_2, D_2) while minimizing oscillations at the end of the transport. Although the oscillations provide a natural way to formulate a cost functional, we use a slightly modified “indirect” approach suggested by Fliess et al. to facilitate a quick comparison. That is, instead of finding a smooth curve $[\tau_0, \tau_f] \ni \tau \mapsto \mathbf{y}(\tau)$ such that $d^r \mathbf{y}/d\tau^r(\tau_0) = d^r \mathbf{y}/d\tau^r(\tau_f) = \mathbf{0}$ for $r = 1, 2, 3, 4$, we choose to minimize,

$$J = \dot{\mathbf{y}}^T(\tau_0) \dot{\mathbf{y}}(\tau_0) + \dot{\mathbf{y}}^T(\tau_f) \dot{\mathbf{y}}(\tau_f) + \ddot{\mathbf{y}}^T(\tau_0) \ddot{\mathbf{y}}(\tau_0) + \ddot{\mathbf{y}}^T(\tau_f) \ddot{\mathbf{y}}(\tau_f) \quad (29)$$

subject to the endpoint constraints,

$$\frac{d^r \mathbf{y}}{d\tau^r}(\tau_0) = \mathbf{0} \quad r = 3, 4 \quad (30)$$

$$\frac{d^r \mathbf{y}}{d\tau^r}(\tau_f) = \mathbf{0} \quad r = 3, 4 \quad (31)$$

The “high-level” control is obtained by [7]

$$D(\tau) = y_1(\tau) - \frac{\dot{y}_1(\tau) y_2(\tau)}{\dot{y}_2(\tau) - g} \quad (32)$$

$$R^2(\tau) = y_2^2(\tau) + \left(\frac{\dot{y}_1(\tau) y_2(\tau)}{\dot{y}_2(\tau) - g} \right)^2 \quad (33)$$

Recall that in our method, the derivatives are obtained by a simple matrix multiplication of the data at the LGL points. Figure 1 displays plots in a form suitable for comparison with Ref. [7]. The number of LGL

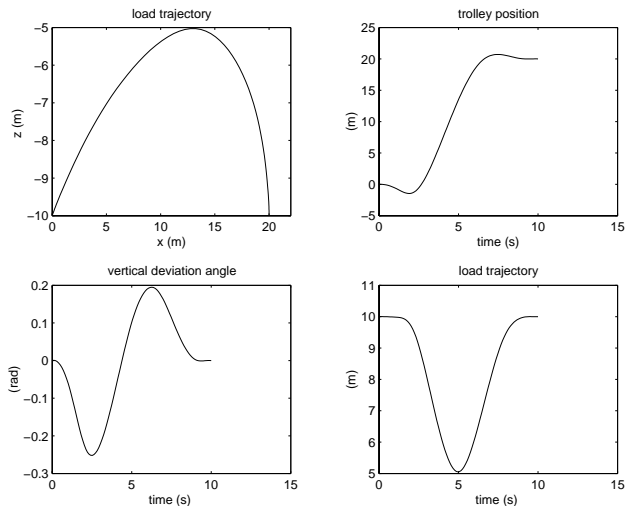


Figure 1: Results for the crane problem

points were arbitrarily chosen to be $N = 11$. Although the shape of our plots is similar to that of Fließ et al., notice that our curves are different. This is because we have chosen to solve the optimal motion planning problem rather than settle for feasible trajectories. Thus, our method generates fewer oscillations (compare the plot of the vertical deviation angle, θ , with that of Ref. [7]). In principle, we can get better results by reformulating the problem as that of either minimizing the oscillations while driving all four derivatives of the output to zero at the boundary points or by keeping the current formulation and impose the additional constraints that the derivatives of θ vanish at the end points. We postpone these ideas for future work.

5 Issues in Using Flat Outputs

While the notion of flatness is a promising idea, it is unclear at this stage whether optimal trajectories should be computed in (the flat) output space. The major property of flatness is the elimination of the system dynamics by endogenous feedback. This comes at some cost. In state-space, the boundary conditions are typically stated simply (e.g. linear boundary conditions) and have physical meaning. The flat output transforms these conditions to a possibly complex (e.g. nonlinear) set of end point conditions (compare Eq.(3) to (9)). The same arguments hold for the transformation of the cost functional. Thus, it is possible that flatness parameterization might actually worsen real-time trajectory optimization. In such situations, a (traditional) change in coordinates might offer faster run times[15]. However, flatness parameterization would still enjoy closed-loop stabilization by endogenous equivalence to linear controllable systems [7].

Method	Intervals	Variables	Cost
NTG*	1	5	1.9127
NTG*	30	65	1.6859
RIOTS*	11		1.7081
RIOTS*	200		1.6857
PS (standard)	4	15	1.8903
PS (standard)	10	33	1.6857
PS (flat)	4	5	1.8903
PS (flat)	10	11	1.6857

* from Ref.[11]

Table 1: Comparison of cost functions from different numerical methods

We demonstrate these issues by considering the forced van der Pol oscillator discussed in [11]. Although this system is static feedback linearizable it illustrates the issues and facilitates a quick comparison to other methods discussed in [11]. The dynamics of this system is given by

$$\dot{x}_1(\tau) = x_2(\tau) \quad (34)$$

$$\dot{x}_2(\tau) = -x_1(\tau) + (1 - x_1^2(\tau))x_2(\tau) + u(\tau) \quad (35)$$

The output $y(\tau) = x_1(\tau)$ is flat since,

$$x_1(\tau) = y(\tau) \quad (36)$$

$$x_2(\tau) = \dot{y}(\tau) \quad (37)$$

$$u = \ddot{y}(\tau) + y(\tau) - (1 - y^2(\tau))\dot{y}(\tau) \quad (38)$$

The optimal control problem is to minimize the quadratic cost function,

$$J = \frac{1}{2} \int_0^5 (x_1^2(\tau) + x_2^2(\tau) + u^2(\tau))d\tau \quad (39)$$

subject to the dynamics given above and the endpoint constraints,

$$x_1(0) = 1, \quad x_2(0) = 0, \quad x_2(5) - x_1(5) = 1$$

Note that the output transforms a numerically well-behaved quadratic integrand to a significantly more complicated function.

Table 1 compares “accuracy” (i.e. value of cost function) to the methods discussed in [11]. PS (standard) implies flatness was not exploited. As in Ref. [11] each test case was run for 100 random initial guesses and the results averaged. From this table it is apparent that both formulations using the PS method yield comparable results in terms of accuracy. Unlike the speed advantage detected in [11] using flat outputs, both PS methods surprisingly yielded comparable run times. However, when we used NPSOL [10], the NLP solver used in [11] (instead of SNOPT [9]), then the flat PS method does run about 10 times faster than the standard PS method. This is at least

in part due to the three-fold difference in the number of variables between the standard and flat PS methods (see Table 1). Apparently, the decrease in the number of optimization variables (and consequent increase in computational speed) offered by flatness parameterization can be overcome. In our experiments, SNOPT was used in dense mode (i.e. the sparsity of the Jacobians was not exploited). Thus, the speed differences are attributable to differences in the implementation of sequential quadratic programming between the two NLP algorithms (such as the use of elastic variables [9]). There are a number of other issues in using flat outputs, some of which are discussed below.

6 Conclusions and Further Work

Pseudospectral methods offer a natural way to solve nonlinear control problems where the dynamics is described in terms of a differential-algebraic state space model. For flat systems, the optimal motion planning problem can be readily solved using PS methods. It is not clear at this stage whether flatness should be exploited in every instance. A change in coordinates or partial inversion might provide faster results. In such situations, it may be advantageous to generate trajectories without full inversion and use the flatness parameterization for stabilization. In all these situations, it is necessary to perform rapid and accurate differentiations. PS methods offer these operations in terms of elementary linear algebra (matrix times vector operations). Much work remains to be done in investigating the interplay between theory and computation. It is possible that a Chebyshev PS method might be better than a Legendre PS method for flat systems since Chebyshev PS methods are based on optimal max-norm interpolation strategy while the Legendre PS method is only L^2 -optimal. Further, higher accuracy for higher-order derivatives are possible by nonlinear time-transformations. Much of these ideas will be explored in future papers, but it is evident that a combination of PS methods with recent advances in nonlinear control theory and optimization can be combined to effectively explore a new way of solving nonlinear control problems.

References

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

[2] Clarke, F. H., *Optimization and Nonsmooth Analysis*, Classics in Applied Mathematics, Vol. 5, SIAM, Philadelphia, PA, 1990.

[3] Elnagar, J., M. A. Kazemi and M. Razzaghi, "The pseudospectral Legendre method for discretiz-

ing optimal control problems," *IEEE Trans. Automat. Contr.*, Vol. 40, No. 10, (1995), pp. 1793-1796.

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

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

[6] Faiz, N., S. K. Agrawal, and R. M. Murray, "Trajectory planning of differentially flat systems with dynamics and inequalities," *J. Guidance, Control and Dynamics*, Vol. 24, No. 2, (2001), pp. 219-227.

[7] Fliess, M., J. Lévine, P. Martin and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *Inter. J. Contr.*, Vol. 61, No. 6, (1995), pp. 1327-1361.

[8] Fliess, M., J. Lévine, P. Martin and P. Rouchon, "A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems," *IEEE Trans. Automat. Contr.*, Vol. 44, No. 5 (1999), pp. 922-937.

[9] Gill, P. E., W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, Numerical Analysis Report 97-2, U.C. San Diego, La Jolla, CA 1997.

[10] Gill, P. E., W. Murray, M. A. Saunders and M. H. Wright, *User's Guide for NPSOL 5.0: A FORTRAN Package for Nonlinear Programming*, Report SOL 86-2, Dept. of Operations Research, Stanford Univ., CA, 1998.

[11] Milam, M. B., K. Mushambi and R. M. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems," *Proc. IEEE Conf. on Decision and Control*, December 2000.

[12] Petit, N., M. B. Milam and R. M. Murray, "Inversion based constrained trajectory optimization," *IFAC-Symp. NOLCOS 2001*.

[13] Ross, I. M., and F. Fahroo, "A pseudospectral transformation of the covectors of optimal control systems," *Proc. IFAC-Symp. System, Structure and Control*, Prague, August 2001.

[14] Ross, I. M., and F. Fahroo, "Convergence of pseudospectral discretizations of optimal control problems," *Proc. IEEE Conf. on Decision and Control*, Orlando, FL, December 2001.

[15] Ross, I. M., J. Rea and F. Fahroo, "Exploiting higher-order derivatives in computational optimal control," *Proc. Med. Conf. Control and Automation*, Lisbon, Portugal, July 2002.

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

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