2019-09

Further Results on Fast Birkhoff Pseudospectral Optimal Control Programming

Ross, I.M.; Proulx, R.J.

American Institute of Aeronautics and Astronautics

Ross, I. M., and R. J. Proulx. "Further Results on Fast Birkhoff Pseudospectral Optimal Control Programming." Journal of Guidance, Control, and Dynamics 42.9 (2019): 2086-2092.
https://hdl.handle.net/10945/64314

JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 42, No. 9, September 2019

Engineering Notes

Further Results on Fast Birkhoff Pseudospectral Optimal Control Programming

I. M. Ross* and R. J. Proulx†
Naval Postgraduate School, Monterey, California 93943

DOI: 10.2514/1.G004297

I. Introduction

IN [1], we introduced a fast mesh-refinement strategy for pseudospectral (PS) optimal control. This strategy was based on Birkhoff PS techniques. A Birkhoff PS method is based on using Birkhoff interpolants [2–4]. A Birkhoff interpolating function generalizes Lagrange and Hermite interpolants [4] by using the values of a heterogeneous mix of various orders of derivatives of a function [3]. The use of suitable Birkhoff interpolants [3,4] offers an emerging PS approach [1,2], in which fast solutions can be generated over a dense grid. As shown in [1], the computational speed of the mesh refinement is very fast [i.e., $\mathcal{O}(1)$].

Unlike Lagrange and Hermite interpolation schemes, several different Birkhoff interpolants may be designed over the same grid. In [1], we presented two viable Birkhoff interpolants that generated two different PS discretizations over the same grid. As in [5,6], the grid in [1] was completely arbitrary. The use of arbitrary grids allows us to present clear ideas so that a proper choice of a particular grid can be made based on the resulting mathematics. In this Note, we follow [1,5,6] by using arbitrary grids for advancing the development of a fast and unified framework for the PS computation of optimal controls. We introduce a virtual optimization variable and show how to unify the two cases presented in [1]. Beyond unification, we also present simplicity in mathematical programming that isolates all the Birkhoff terms to a linear system; as a result, all of the optimal control data functions can then be handled separately and without regard to the specificity of the grid points. The grid-agnostic programming method allows us to rapidly test the performance of various grid distributions. The results for Lobatto, Radau, and pure Gauss-type grids for both Legendre—and Chebyshev–Birkhoff PS methods are presented for a benchmark singular optimal control problem [7–9]. The preliminary results indicate that a Birkhoff PS method is remarkably robust in the sense that it is able to solve for singular arcs across 12 different implementations associated with six different Gaussian distributions. The mathematics of the virtual optimization variable, the resulting unified implementation details, the generation of efficient computational formulas for the Birkhoff data functions, and the demonstration of the numerical robustness of the Birkhoff PS method across a plethora of different Gaussian grids for a singular optimal control problem are the subjects of the Note.

Received 23 December 2018; revision received 4 March 2019; accepted for publication 7 March 2019; published online 29 April 2019. Copyright © 2019 by I. M. Ross and R. J. Proulx. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-3884 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Distinguished Professor and Program Director, Control and Optimization, Department of Mechanical and Aerospace Engineering.

†Research Professor, Control and Optimization Laboratories, Space Systems Academic Group.

II. Brief Review of Birkhoff PS Optimal Control Theory

One surprising aspect of the PS theory is that the discussions become simple and more elegant when an arbitrary grid is used to illustrate its features. This is, in part, because the use of a specific grid at the outset tends to obfuscate an otherwise clear idea. To this end, we set

$$\pi^N := [\tau_0, \tau_1, \ldots, \tau_N] \tag{1}$$

to be an arbitrary grid (or "mesh") of points (see Fig. 1), such that

$$-\infty < \tau^0 \le \tau_0 < \tau_1 < \cdots < \tau_{N-1} < \tau_N \le \tau^f < \infty \tag{2}$$

Following [1], we define two subsets of the grid π^N given by

$$\pi_a^N := [\tau_1, \ldots, \tau_N] \tag{3a}$$

$$\pi_b^N := [\tau_0, \tau_1, \ldots, \tau_{N-1}] \tag{3b}$$

Thus, π^N may be represented either by $\pi^N = [\tau_0, \pi_a^N]$ or $\pi^N = [\pi_b^N, \tau_N]$.

Let $\mathbb{R} \ni \tau \mapsto y \in \mathbb{R}$ be a continuous bounded function with bounded derivatives, as illustrated in Fig. 1. Then, two first-order Birkhoff interpolants can be defined as [1]

$$I_a^N y(\tau) := y(\tau_0)B_0^a(\tau) + \sum_{j=1}^{N} \dot{y}(\tau_j)B_j^a(\tau) \tag{4a}$$

$$I_b^N y(\tau) := \sum_{j=0}^{N-1} \dot{y}(\tau_j)B_j^b(\tau) + y(\tau_N)B_N^b(\tau) \tag{4b}$$

in which I_θ^N, $\theta \in \{a, b\}$ are the two interpolation operators, and B_j^θ, $\theta \in \{a, b\}$, $j = 0, 1, \ldots, N$ are (Birkhoff) functions that satisfy the interpolation conditions

$$\begin{aligned} B_0^a(\tau_0) &= 1, & B_j^a(\tau_0) &= 0, & j &= 1, \ldots, N \\ \dot{B}_0^a(\tau_i) &= 0, & \dot{B}_j^a(\tau_i) &= \delta_{ij}, & i &= 1, \ldots, N \end{aligned} \tag{5a}$$

$$\begin{aligned} B_N^b(\tau_N) &= 1, & B_j^b(\tau_N) &= 0, & j &= 0, \ldots, N-1 \\ \dot{B}_N^b(\tau_i) &= 0, & \dot{B}_j^b(\tau_i) &= \delta_{ij}, & i &= 0, \ldots, N-1 \end{aligned} \tag{5b}$$

The quantity δ_{ij} in Eq. (5) is the Kronecker delta.

Remark 1: Equation (5) is obtained by simply imposing the interpolation conditions:

$$I_a^N y(\tau_0) = y(\tau_0), \qquad \frac{d}{d\tau}(I_a^N y(\tau))\Big|_{\tau=\tau_j} = \dot{y}(\tau_i), \qquad i = 1, \ldots, N \tag{6a}$$

$$I_b^N y(\tau_N) = y(\tau_N), \qquad \frac{d}{d\tau}(I_b^N y(\tau))\Big|_{\tau=\tau_j} = \dot{y}(\tau_i), \qquad i = 0, \ldots, N-1 \tag{6b}$$

Remark 2: Although we will eventually limit the scope of our computations to Birkhoff basis polynomials [i.e., polynomials that

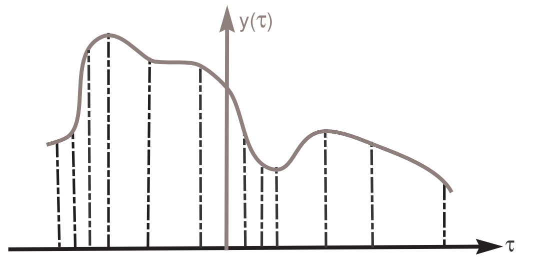

Fig. 1 Illustration of an arbitrary grid in a generic PS method [6].

satisfy Eq. (5)], note that there is no assumption of polynomials in all of the preceding equations.

Deferring a discussion of the details associated with vector-valued functions to Sec. III.B, we consider scalar-valued functions in the optimal control problem given by

$$x \in \mathbb{R}, \qquad u \in \mathbb{R}, \qquad \tau \in [\tau^0, \tau^f]$$

$$(P)\begin{cases} \text{minimize} & J[x(\cdot), u(\cdot)] = E(x(\tau^0), x(\tau^f)) \\ \text{subject to} & \dot{x}(\tau) = f(x(\tau), u(\tau)) \\ & e(x(\tau^0), x(\tau^f)) = 0 \end{cases} \tag{7}$$

Consider the application of Eq. (4a) to discretize problem P. To achieve this goal, we define

$$x^N(\tau) := x_0 B_0^a(\tau) + \sum_{j=1}^{N} v_j B_j^a(\tau) \tag{8}$$

in which x_0 and v_j, $j = 1, \ldots, N$ are the unknown optimization variables over the arbitrary grid π^N, with $\tau_0 := \tau^0$ and $\tau_N := \tau^f$. Differentiating both sides of Eq. (8) and substituting the result in the dynamic constraint $\dot{x}^N(\tau) = f(x^N(\tau), u(\tau))$, we get

$$x_0 \dot{B}_0^a(\tau) + \sum_{j=1}^{N} v_j \dot{B}_j^a(\tau) = f(x^N(\tau), u(\tau)) \tag{9}$$

Evaluating Eq. (9) over the grid $\pi^N = [\tau_0, \pi_a^N]$ and using the result to approximate the dynamics, we get a candidate discretization of problem P given by [1]

$$X = (x_0, X_a) \in \mathbb{R}^{N+1}, \qquad U = (u_0, U_a) \in \mathbb{R}^{N+1}, \qquad V_a \in \mathbb{R}^N$$

$$(P_a^N)\begin{cases} \text{minimize} & J_a^N[X, U, V_a] := E(x_0, x_N) \\ \text{subject to} & V_a = f(X_a, U_a) \\ & X_a = x_0 b_0 + \boldsymbol{B}_a V_a \\ & I_a V_a = f(x_0, u_0) - x_0 \dot{B}_0(\tau_0) \\ & e(x_0, x_N) = 0 \end{cases} \tag{10}$$

in which

$$X_a := (x^N(\tau_1), \ldots, x^N(\tau_N)) \;\; U_a := (u(\tau_1), \ldots, u(\tau_N)) \;\; V_a := (v_1, \ldots, v_N) \tag{11a}$$

$$I_a := (\dot{B}_1^a(\tau_0), \ldots, \dot{B}_N^a(\tau_0))^T \quad b_0 := (B_0^a(\tau_1), \ldots, B_0^a(\tau_N))$$
$$\boldsymbol{B}_a := (B_j^a(\tau_i))_{1 \le i,j \le N} \tag{11b}$$

and f is reused as an overloaded function (see [8] p. 8) defined by

$$f(X_a, U_a) := (f(x_1, u_1), \ldots, f(x_N, u_N))$$

Remark 3: Only the state trajectory $x(\cdot)$ is approximated in terms of Birkhoff basis functions. No specific assumption has been made on the approximation of the control trajectory $u(\cdot)$. The value of the control encoded in U is simply the sampled values of any function that renders dynamic feasibility [8] at each point over the arbitrary grid π^N.

If we use Eq. (4b) to discretize problem P and follow the same process as in arriving at Eq. (10), we get

$$X := (X_b, x_N) \in \mathbb{R}^{N+1}, \qquad U := (U_b, u_N) \in \mathbb{R}^{N+1}, \qquad V_b \in \mathbb{R}^N$$

$$(P_b^N)\begin{cases} \text{minimize} & J_b^N[X, U, V_b] := E(x_0, x_N) \\ \text{subject to} & V_b = f(X_b, U_b) \\ & X_b = \boldsymbol{B}_b V_b + x_N b_N \\ & I_b V_b = f(x_N, u_N) - x_N \dot{B}_N(\tau_N) \\ & e(x_0, x_N) = 0 \end{cases} \tag{12}$$

in which

$$X_b := (x^N(\tau_0), \ldots, x^N(\tau_{N-1})) \;\; U_b := (u(\tau_0), \ldots, u(\tau_{N-1}))$$
$$V_b := (v_0, \ldots, v_{N-1}) \tag{13a}$$

$$I_b := (\dot{B}_1^b(\tau_N), \ldots, \dot{B}_N^b(\tau_N))^T \qquad b_N := (B_N^b(\tau_0), \ldots, B_N^b(\tau_{N-1}))$$
$$\boldsymbol{B}_b := (B_j^b(\tau_i))_{0 \le i,j \le N-1} \tag{13b}$$

III. Unified Mathematical Programming Problem Formulation

As noted in [1], $B_0^a(\tau) = 1 = B_N^b(\tau)$ for all $\tau \in [\tau^0, \tau^f]$. This essentially follows from Eq. (4) and by considering the case $y(\tau) = \text{constant}$. Consequently, we have

$$\dot{B}_0^a(\tau) = 0 \tag{14a}$$

$$\dot{B}_N^b(\tau) = 0 \tag{14b}$$

As a result, both b_0 and b_N are simply equal to an $\boldsymbol{N}$ vector of ones. Denoting this vector as $\mathbf{1}$, it is possible to produce a unified mathematical programming problem for all cases after the introduction of new variables and some rearrangement of the resulting equations.

A. Reformulation of the Basic Problems

Define scalars v_0 and v_N for problems P_a^N and P_b^N, respectively, according to

$$v_0 := I_a V_a \tag{15a}$$

$$v_N := I_b V_b \tag{15b}$$

Setting $V := [v_0, v_1, \ldots, v_N]^T$ as an "augmented" optimization variable, Eqs. (10) and (12) can be rewritten as

$$X \in \mathbb{R}^{N+1}, \quad U \in \mathbb{R}^{N+1}, \quad V \in \mathbb{R}^{N+1}$$

$$(P_a^N)\begin{cases} \text{minimize} & J_a^N[X, U, V] := E(x_0, x_N) \\ \text{subject to} & V = f(X, U) \\ & X_a = x_0 b_0 + \boldsymbol{B}_a V_a \\ & I_a V_a = v_0 \\ & e(x_0, x_N) = 0 \end{cases} \tag{16a}$$

$$X \in \mathbb{R}^{N+1}, \quad U \in \mathbb{R}^{N+1}, \quad V \in \mathbb{R}^{N+1}$$

$$(P_b^N)\begin{cases} \text{minimize} & J_b^N[X,U,V] := E(x_0, x_N) \\ \text{subject to} & V = f(X,U) \\ & X_b = \boldsymbol{B}_b V_b + x_N b_N \\ & I_b V_b = v_N \\ & e(x_0, x_N) = 0 \end{cases} \tag{16b}$$

Let $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$ be two matrices defined according to

$$\boldsymbol{A}_a := \left[\begin{array}{cc:cc} \mathbf{1} & -\boldsymbol{I}_N & \mathbf{0} & \boldsymbol{B}_a \\ 0 & \mathbf{0}^T & 1 & -I_a \end{array}\right] \tag{17a}$$

$$\boldsymbol{A}_b := \left[\begin{array}{cc:cc} -\boldsymbol{I}_N & \mathbf{1} & \boldsymbol{B}_b & \mathbf{0} \\ \mathbf{0}^T & 0 & -I_b & 1 \end{array}\right] \tag{17b}$$

Then, a unified discretization of problem P may now be written as

$$X \in \mathbb{R}^{N+1}, \quad U \in \mathbb{R}^{N+1}, \quad V \in \mathbb{R}^{N+1}$$

$$(P_\theta^N)\begin{cases} \text{minimize} & J^N[X,U,V] := E(x_0, x_N) \\ \text{subject to} & A_\theta \begin{bmatrix} X \\ V \end{bmatrix} = \mathbf{0} \\ & V = f(X,U) \\ & e(x_0, x_N) = 0 \end{cases} \tag{18}$$

An extension of the unification process to vector-valued data functions is theoretically straightforward; however, we achieve new insights by way of a specific organization of the variables, as shown next.

B. Structured Mathematical Programming for the General Problem

Consider a generic optimal control problem given by

$$\boldsymbol{x} \in \mathbb{R}^{N_x}, \qquad \boldsymbol{u} \in \mathbb{R}^{N_u}, \qquad t \in [t_0, t_f]$$

$$(\boldsymbol{P})\begin{cases} \text{minimize} & J[\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot), t_0, t_f] = E(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) \\ \text{subject to} & d\boldsymbol{x}(t)/dt = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ & \boldsymbol{e}(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) = \mathbf{0} \end{cases} \tag{19}$$

in which we now allow $N_x > 1$, $N_u > 1$, $\boldsymbol{e}: \mathbb{R}^{N_x} \times \mathbb{R}^{N_x} \to \mathbb{R}^{N_e}$, with $N_e \geq 1$, and the horizon $t_f - t_0 > 0$ also being an optimization variable in addition to the clock times t_0 and t_f. The incorporation of inequalities and path constraints in Eq. (19) is straightforward; we deliberately avoid these constraints for brevity. The generation of an efficient mathematical programming formulation of a discretization of problem $\boldsymbol{P}$ involves three key steps: 1) a selection of an appropriate function for the domain transformation, 2) a discriminating matrix organization of the discretized vectors, and 3) a matrix–vector mathematical programming problem formulation.

1. *Domain Transformation*

In following [8,10], we first perform a domain transformation using a suitable function Γ:

$$t = \Gamma(\tau, t_0, t_f; \boldsymbol{p}) \tag{20}$$

in which $\boldsymbol{p} \in \mathbb{R}^{N_p}$ is a parameter that can be selected or optimized [10,11], and Γ is such that $t_0 = \Gamma(\tau^0, t_0, t_f; \boldsymbol{p})$ and $t_f = \Gamma(\tau^f, t_0, t_f; \boldsymbol{p})$. With $\tau^f = 1 = -\tau^0$, the simplest choice of Γ for a finite horizon problem is the affine function:

$$t = \Gamma(\tau, t_0, t_f) := \left(\frac{t_f - t_0}{2}\right)\tau + \left(\frac{t_f + t_0}{2}\right) \tag{21}$$

For infinite horizon problems (i.e., if $[t_0, t_f]$ is replaced by $[0, \infty)$), a common choice [8,12,13] for Γ is

$$t = \Gamma(\tau; p) := p\left(\frac{1+\tau}{1-\tau}\right), \qquad p > 0 \tag{22}$$

in which $\tau \in [-1, 1]$. Thus, with a suitable choice of Γ, we transform the differential equations according to

$$\dot{\boldsymbol{x}}(\tau) = \left(\frac{d\Gamma}{d\tau}\right)\boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \tag{23}$$

In Eq. (23), we have abused notation for expediency. The symbol $\boldsymbol{x}(\tau)$ should be written more appropriately as $\boldsymbol{x}(\Gamma(\tau, t_0, t_f; \boldsymbol{p}))$. We have avoided such elaborations for simplicity of notation for $\boldsymbol{u}(\cdot)$, $\dot{\boldsymbol{x}}(\cdot)$ and $d\Gamma/d\tau$ as well.

Remark 4: For finite horizon problems, spectral efficiency can be enhanced by using a nonlinear time domain transformation $\tau \mapsto t$ instead of the affine formula given by Eq. (21). Indeed, in DIDO [8,14], a state-of-the-art MATLAB® optimal control toolbox, a nonlinear time domain transformation $[-1, 1] \mapsto [t_0, t_f]$ is used to support anti-aliasing [15,16] and assist in the resolution of the system trajectory [8,11,16,17].

2. *Matrix Organization of the Discretized Vectors*

Using Eq. (23), we can construct a version of problem P_θ^N for problem $\boldsymbol{P}$ by repeating the process of the previous subsection to each variable. Although this is theoretically viable, it is practically unsatisfactory. To generate an efficient multidimensional version of problem P_θ^N that is symbolically connected to problem $\boldsymbol{P}$ in some desirable manner, we define three matrices according to

$$\boldsymbol{X} := [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \qquad \in \mathbb{R}^{N_x \times (N+1)} \tag{24}$$

$$\boldsymbol{V} := [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N] \qquad \in \mathbb{R}^{N_x \times (N+1)} \tag{25}$$

$$\boldsymbol{U} := [\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N] \qquad \in \mathbb{R}^{N_u \times (N+1)} \tag{26}$$

in which $\boldsymbol{x}_k \in \mathbb{R}^{N_x}$, $\boldsymbol{v}_k \in \mathbb{R}^{N_x}$, $\boldsymbol{u}_k \in \mathbb{R}^{N_u}$, $k = 0, 1, \ldots, N$.

Remark 5: The matrix organization of the discretized optimal control variables given by Eqs. (24–26) was first introduced in the year 2001 as part of a MATLAB programming technique in DIDO [8]. The variables $\boldsymbol{X}$ and $\boldsymbol{U}$ are packed in a single primal structure as primal.**states** and primal.**controls**, respectively. The corresponding domain transformed grid, π^N, is contained in the same primal structure under primal.**time**. Such DIDO programming techniques are now quite popular and replicated in many other software packages. See pp. ix–x and Sec. 1.1.2 in [8] for further details.

3. *Matrix–Vector Mathematical Programming Problem Formulation*

Using $\boldsymbol{X}$, $\boldsymbol{V}$, and $\boldsymbol{U}$ as matrices of optimization variables, it follows that a Birkhoff discretization of problem $\boldsymbol{P}$ can be framed as

$$\boldsymbol{X} \in \mathbb{R}^{N_x \times (N+1)}, \quad \boldsymbol{U} \in \mathbb{R}^{N_u \times (N+1)}, \quad \boldsymbol{V} \in \mathbb{R}^{N_x \times (N+1)}, \quad t_0 \in \mathbb{R}, \quad t_f \in \mathbb{R}$$

$$(\boldsymbol{P}_\theta^N)\begin{cases} \text{minimize} & J^N[\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{V}, t_0, t_f] := E(\boldsymbol{x}_0, \boldsymbol{x}_N, t_0, t_f) \\ \text{subject to} & A_\theta \begin{bmatrix} \boldsymbol{X}^T \\ \boldsymbol{V}^T \end{bmatrix} = \mathbf{0} \\ & \boldsymbol{V} = \left(\frac{d\Gamma}{d\tau}\right)\boldsymbol{f}(\boldsymbol{X}, \boldsymbol{U}) \\ & \boldsymbol{e}(\boldsymbol{x}_0, \boldsymbol{x}_N, t_0, t_f) = \mathbf{0} \end{cases} \tag{27}$$

in which we have reused $\boldsymbol{f}$ as an overloaded operator (see [8] p. 8) defined by

$$\boldsymbol{f}(\boldsymbol{X},\boldsymbol{U}) := [\boldsymbol{f}(\boldsymbol{x}_0,\boldsymbol{u}_0),\boldsymbol{f}(\boldsymbol{x}_1,\boldsymbol{u}_1),\,\ldots,\boldsymbol{f}(\boldsymbol{x}_N,\boldsymbol{u}_N)] \in \mathbb{R}^{N_x\times(N+1)}$$

Remark 6: Because matrix properties are used to define it, problem $\boldsymbol{P}_\theta^N$ is not a nonlinear programming (NLP) problem in a standard form, in which the optimization variables are organized in terms of a single vector. Note also that the constraint equations in Eq. (27) are not described in terms of vectors; rather, they are given compactly in terms of matrices and functions of matrices.

Remark 7: By inspection of Eq. (27), it follows that the number of optimization variables grows as $\mathcal{O}(N)$; the constant growth rate c is given by $c = (2N_x + N_u)$. In its matrix form, problem $\boldsymbol{P}_\theta^N$ is dense; however, it can be transformed to a sparse NLP problem by rearranging all the matrix variables to a single column vector. Under this construct, the density of the (Jacobian of the) transformed data functions varies as $\mathcal{O}(1/N)$. The constant density drop rate is given by $c = (2N_x + N_u)$.

From Remarks 6 and 7, it follows that the "space" of discretized optimal control problems is much smaller than the space of NLP. In other words, not all NLP originate from transformations of discretized optimal control problems. Consequently, it is inadvisable to simply patch a discretized optimal control problem to an NLP solver. Although such naive methods are quite popular, the resulting minimalist's approach severely limits the capability of an otherwise more powerful idea [14]. To solve problem $\boldsymbol{P}_\theta^N$ effectively, an optimal-control-centric algorithm is needed. The first step in this direction was taken in [18] in the form of a spectral algorithm. An advanced, guess-free version of this algorithm is implemented in DIDO. This algorithm can solve many reportedly hard problems [19] with embarrassing ease; see [8,20] for details. Note also that it is possible to use the optimal control theory itself to construct new algorithms [21].

IV. Efficient Computational Formulas for the Birkhoff Matrices

A fast optimal control programming algorithm mandates an efficient computation of the constituent elements of problem $\boldsymbol{P}_\theta^N$. A quick examination of Eq. (27) shows that all the Birkhoff terms are isolated and packed in the matrix $\boldsymbol{A}_\theta$. The remainder of the equations are problem specific. Hence, a fast computation of $\boldsymbol{A}_\theta$ supports performance gains across all problems. To this end, we consider the problem of finding functions that satisfy Eq. (5). We select the space of polynomials for these basis functions. These are the Birkhoff basis polynomials [2]. In much the same way as it is convenient to express the components of a finite-dimensional vector by the use of an orthogonal coordinate system, significant efficiencies can be obtained by expressing a Birkhoff polynomial in terms of orthogonal polynomials. To better understand the best choice of orthogonal polynomials, we begin with a generic set of orthogonal polynomials $p_k(\tau), k = 0, 1, \ldots, N, \tau \in [-1, 1]$ of degree at most N that span $\mathcal{P}^N$, the space of polynomials of degree N.

From the assumption of orthogonality, we have

$$\int_{-1}^{1} p_n(\tau)p_m(\tau)\omega(\tau)\,\mathrm{d}\tau \begin{cases} = 0, & \text{if } n \neq m \\ > 0, & \text{if } n = m \end{cases} \tag{28}$$

in which $\omega(\tau)$ is some nonnegative weight function. Setting

$$\gamma_k := \int_{-1}^{1} p_k(\tau)p_k(\tau)\omega(\tau)\,\mathrm{d}\tau \tag{29}$$

it follows that any polynomial $q(\cdot) \in \mathcal{P}^N$ may be expressed in terms of its orthogonal components

$$q(\tau) = \sum_{k=0}^{N} c_k\left(\frac{p_k(\tau)}{\gamma_k}\right) \tag{30}$$

in which $c_k \in \mathbb{R}$ are the components of $q(\cdot)$ along the orthogonal directions, $p_k, k = 0, 1, \ldots, N$. The coefficient c_k can be computed by taking dot products on both sides of Eq. (30) according to

$$\int_{-1}^{1} q(\tau)p_l(\tau)\omega(\tau)\,\mathrm{d}\tau = \sum_{k=0}^{N} c_k \int_{-1}^{1}\left(\frac{p_k(\tau)p_l(\tau)\omega(\tau)}{\gamma_k}\right)\mathrm{d}\tau = c_l, \quad l = 0, 1, \ldots, N \tag{31}$$

A. Explicit Computational Formulas for $\boldsymbol{B}^\theta$

A quick examination of Eq. (5) shows that it is prudent to find the derivative of Birkhoff polynomials and integrate it afterward; hence, we follow [2] and use Eq. (30) to express $\dot{B}_j^\theta(\tau)$ as a polynomial of degree $N - 1$:

$$\dot{B}_j^\theta(\tau) = \sum_{k=0}^{N-1} \alpha_{kj}^\theta\left(\frac{p_k(\tau)}{\gamma_k}\right) \tag{32}$$

in which $\alpha_{kj}^\theta \in \mathbb{R}$ are the $k = 0, 1, \ldots, N - 1$ unknown components of $\dot{B}_j^\theta(\cdot)$ for each $j = 0, 1, \ldots, N$. Integrating both sides of Eq. (32), we can write

$$\int_{\tau_0}^{\tau} \dot{B}_j^a(\xi)\,\mathrm{d}\xi = B_j^a(\tau) - B_j^a(\tau_0) = \sum_{k=0}^{N-1}\left(\frac{\alpha_{kj}^a}{\gamma_k}\right)\int_{\tau_0}^{\tau} p_k(\xi)\,\mathrm{d}\xi \tag{33a}$$

$$\int_{\tau}^{\tau_N} \dot{B}_j^b(\xi)\,\mathrm{d}\xi = B_j^b(\tau_N) - B_j^b(\tau) = \sum_{k=0}^{N-1}\left(\frac{\alpha_{kj}^b}{\gamma_k}\right)\int_{\tau}^{\tau_N} p_k(\xi)\,\mathrm{d}\xi \tag{33b}$$

As noted in [2], the boundary conditions given in Eq. (5) simplify Eq. (33) to

$$B_j^a(\tau) = \sum_{k=0}^{N-1}\left(\frac{\alpha_{kj}^a}{\gamma_k}\right)\int_{\tau_0}^{\tau} p_k(\xi)\,\mathrm{d}\xi, \qquad j = 1, \ldots, N \tag{34a}$$

$$B_j^b(\tau) = \sum_{k=0}^{N-1}\left(\frac{\alpha_{kj}^b}{\gamma_k}\right)\int_{\tau_N}^{\tau} p_k(\xi)\,\mathrm{d}\xi \qquad j = 0, \ldots, N - 1 \tag{34b}$$

Thus, the computation of Birkhoff polynomials now reduces to an efficient computation of just the following three quantities: 1) α_{kj}^θ, 2) $\int_{\tau_0}^{\tau} p_k(\xi)\,\mathrm{d}\xi$ and $\int_{\tau_N}^{\tau} p_k(\xi)\,\mathrm{d}\xi$, and 3) γ_k.

B. Computation of α_{kj}^θ

Analogous to Eq. (31), we can compute α_{kj}^θ by taking the dot product on both sides of Eq. (32); this generates

$$\alpha_{lj}^\theta = \int_{-1}^{1} \dot{B}_j^\theta(\tau)p_l(\tau)\omega(\tau)\,\mathrm{d}\tau, \qquad l = 0, 1, \ldots, N - 1 \tag{35}$$

The integrand in Eq. (35) is a polynomial of degree at most $(2N - 2)$; hence, using an appropriate Gaussian quadrature formula [22–24], we can write

$$\alpha_{lj}^\theta = \sum_{n=0}^{N} \dot{B}_j^\theta(\tau_n)p_l(\tau_n)w_n, \qquad l = 0, 1, \ldots, N - 1 \tag{36}$$

in which $\tau_n, n = 0, \ldots, N$ are $(N + 1)$ quadrature points that are, in principle, independent of the points in π^N. If this choice of independence is made, then we will be unable to efficiently exploit the Kronecker conditions stipulated in Eq. (5). For this reason, we now choose the grid $\pi^N \equiv \pi_G^N$ to be based on Gaussian quadrature formulas that allow us to set:

$$\dot{B}_j^\theta(\tau_n) = \delta_{jn}^\theta = \begin{cases} \delta_{jn}, 1 \leq j, n \leq N, & \text{if } \theta = a \\ \delta_{jn}, 0 \leq j, n \leq N - 1, & \text{if } \theta = b \end{cases} \tag{37}$$

Remark 8: All equations before Eq. (36) are valid for a completely arbitrary grid. The juxtaposition of Eqs. (36) and (37) explains why we choose Gaussian grids.

Limiting our discussions henceforth to Gaussian grids, we get

$$\alpha^a_{lj} = \dot{B}^a_j(\tau_0)p_l(\tau_0)w_0 + p_l(\tau_j)w_j \quad j = 1, \ldots, N, \quad l = 0, \ldots, N-1 \tag{38a}$$

$$\alpha^b_{lj} = p_l(\tau_j)w_j + \dot{B}^b_j(\tau_N)p_l(\tau_N)w_N \quad j = 0, \ldots, N-1, \; l = 0, \ldots, N-1 \tag{38b}$$

in which $\tau_0, \tau_1, \ldots, \tau_N$ are Gaussian points associated with the choice of the orthogonal polynomials.

To determine $\dot{B}^a_j(\tau_0)$ and $\dot{B}^b_j(\tau_N)$ in Eq. (38), we use the fact that $\dot{B}^\theta_j(\cdot) \in \mathcal{P}^{N-1}$; hence, we have the orthogonality condition:

$$\int_{-1}^{1} \dot{B}^\theta_j(\tau)p_N(\tau)\omega(\tau)\,\mathrm{d}\tau = 0, \qquad j = 0, 1, \ldots, N \tag{39}$$

in which p_N is the orthogonal polynomial of degree N. Using the same Gaussian grid π^N_G as before, Eq. (39) can be written as

$$\sum_{n=0}^{N} \dot{B}^\theta_j(\tau_n)p_N(\tau_n)w_n = 0 \qquad j = 0, 1, \ldots, N \tag{40}$$

Using Eq. (5), Eq. (40) simplifies to

$$\dot{B}^a_j(\tau_0)p_N(\tau_0)w_0 + p_N(\tau_j)w_j = 0 \tag{41a}$$

$$p_N(\tau_j)w_j + \dot{B}^b_j(\tau_N)p_N(\tau_N)w_N = 0 \tag{41b}$$

for $j = 0, 1, \ldots, N$.

Collecting all relevant equations and, in particular, the results from Eqs. (14), (32), (38), and (41), we arrive at the following simple, fast, and efficient computational formulas for α^θ_{kj}:

$$\alpha^a_{k0} = 0 \qquad \alpha^a_{kj} = w_j\left(p_k(\tau_j) - \frac{p_N(\tau_j)p_k(\tau_0)}{p_N(\tau_0)}\right) \qquad j = 1, \ldots, N \tag{42a}$$

$$\alpha^b_{kN} = 0 \quad \alpha^b_{kj} = w_j\left(p_k(\tau_j) - \frac{p_N(\tau_j)p_k(\tau_N)}{p_N(\tau_N)}\right) \quad j = 0, \ldots, N-1 \tag{42b}$$

for $k = 0, 1, \ldots, N-1$.

Remark 9: Equation (41) also provides a computational formula for constructing I_a and I_b.

Remark 10: The computational formulas given in Eq. (42) apply to all choices of orthogonal polynomials.

C. Computation of $\int_{\tau_0}^{\tau} p_k(\xi)\,\mathrm{d}\xi$ and $\int_{\tau_N}^{\tau} p_k(\xi)\,\mathrm{d}\xi$

An orthogonal polynomial can be expressed in terms of a linear combination of its derivatives [25]. Hence, computing its integrals may be reduced to an evaluation of a linear combination of the orthogonal polynomials. In limiting the scope of this Note, we demonstrate the production of efficient computational formulas for just the "big two" Gegenbauer polynomials, namely, Legendre and Chebyshev [22].

1. *Legendre: $p_k = P_k$*

If the generic orthogonal polynomial p_k is chosen to be a Legendre polynomial denoted by P_k, we have [23]

$$P_k(\tau) = \frac{1}{2k+1}(\dot{P}_{k+1}(\tau) - \dot{P}_{k-1}(\tau)), \qquad k \geq 1 \tag{43}$$

Integrating both sides of Eq. (43), we get

$$\int_{\tau_0}^{\tau} P_k(\tau)\,\mathrm{d}\tau = \frac{1}{2k+1}(P_{k+1}(\tau) - P_{k-1}(\tau) + P_{k-1}(\tau_0) - P_{k+1}(\tau_0)), \qquad k \geq 1 \tag{44a}$$

$$\int_{\tau_N}^{\tau} P_k(\tau)\,\mathrm{d}\tau = \frac{1}{2k+1}(P_{k+1}(\tau) - P_{k-1}(\tau) + P_{k-1}(\tau_N) - P_{k+1}(\tau_N)), \qquad k \geq 1 \tag{44b}$$

2. *Chebyshev: $p_k = T_k$*

If p_k is chosen to the Chebyshev polynomial of the first kind, T_k, we have [22,23]

$$T_k(\tau) = \frac{\dot{T}_{k+1}(\tau)}{2(k+1)} - \frac{\dot{T}_{k-1}(\tau)}{2(k-1)}, \qquad k \geq 2 \tag{45}$$

Integrating both sides of Eq. (45), we get

$$\int_{\tau_0}^{\tau} T_k(\tau)\,\mathrm{d}\tau = \frac{1}{2(k+1)}(T_{k+1}(\tau) - T_{k+1}(\tau_0)) - \frac{1}{2(k-1)}(T_{k-1}(\tau) - T_{k-1}(\tau_0)), \qquad k \geq 1 \tag{46a}$$

$$\int_{\tau_N}^{\tau} T_k(\tau)\,\mathrm{d}\tau = \frac{1}{2(k+1)}(T_{k+1}(\tau) - T_{k+1}(\tau_N)) - \frac{1}{2(k-1)}(T_{k-1}(\tau) - T_{k-1}(\tau_N)), \qquad k \geq 1 \tag{46b}$$

D. Computation of γ_k

The quantity γ_k is given in closed form according to

$$\text{Legendre: } \gamma_k = \frac{2}{2k+1} \tag{47}$$

$$\text{Chebyshev: } \gamma_k = \begin{cases} \pi, & \text{if } k = 0 \\ \pi/2, & \text{if } k \geq 1 \end{cases} \tag{48}$$

Remark 11: The computation of γ_k and the integrals described in Sec. IV.D do not depend upon the choice of grid associated with the polynomial of choice. Consequently, the formulas provided apply to all types of Legendre and Chebyshev grids.

V. Numerical Tests over a Singular Optimal Control Problem

The Gong problem [7–9] is a deceptively simple singular optimal control problem given by

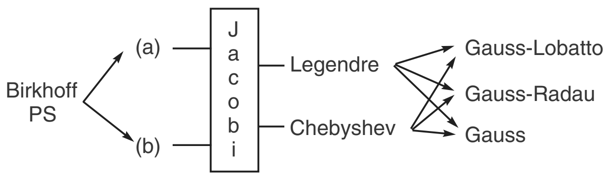

Fig. 2 Schematic for generating at least 12 variants of the Birkhoff PS method.

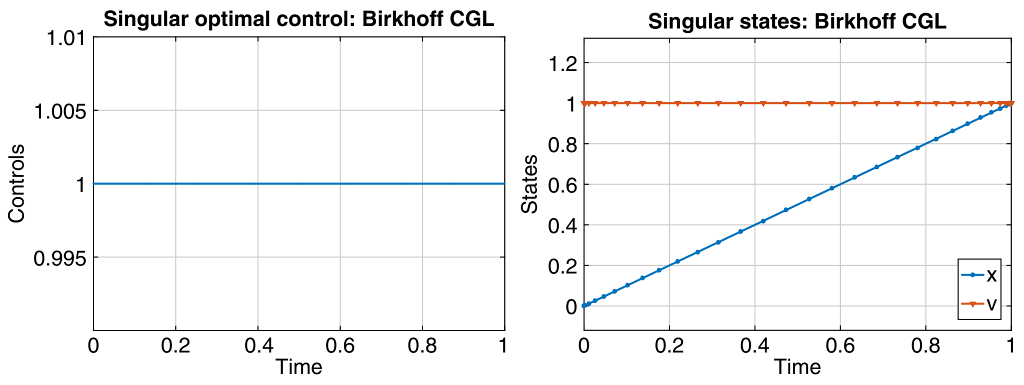

Fig. 3 Singular arc via a Birkhoff PS method; see Fig. 2.

$$\boldsymbol{x} := (x, v) \in \mathbb{R}^2, \qquad u \in \mathbb{U} := \{u \in \mathbb{R}: 0 \le u \le 2\}, \qquad t \in [0, 1]$$

$$(G)\begin{cases} \text{minimize} & J[\boldsymbol{x}(\cdot), u(\cdot)] = \int_0^1 v(t)u(t)\,\mathrm{d}t \\ \text{subject to} & \dot{x}(t) = v(t) \\ & \dot{v}(t) = -v(t) + u(t) \\ & (x(0), v(0)) = (0, 1) \\ & (x(1), v(1)) = (1, 1) \end{cases} \tag{49}$$

The problem is singular because $\partial^2 H/\partial u^2 = 0$, in which H is the Pontryagin Hamiltonian [8]. This problem has previously served well as an excellent test case for discriminating different discretization schemes [8,9] and grid selections [7,8]. In fact, a vast number of computational methods fail to solve this problem even though the exact solution is very simple:

$$x(t) = t \qquad v(t) = 1 \qquad u(t) = 1 \tag{50}$$

Typically, low-order computational methods (e.g., Runge–Kutta) or improper implementations of PS methods fail to find a solution to this problem [7,9]. This is not entirely surprising because low-order computational methods require index-reduction transformations to solve singular optimal control problems [19]. In sharp contrast, a correct implementation of a PS method can find solutions to a broad class of singular optimal control problems without going through a laborious process of index reduction; see [20] for a detailed comparative analysis. For all these reasons, the Gong problem serves an excellent simple test case to explore the correctness of a Birkhoff PS method.

Our unified mathematical programming formulation presented in Sec. III allows us to quickly test a plethora of variants of the Birkhoff PS method by way of different choices of orthogonal functions and grid sections. Figure 2 shows how to generate at least 12 variants of the Birkhoff PS method using the formulas presented in Sec. IV. An implementation of all 12 variants of the Birkhoff PS method generated virtually identical solutions to the Gong problem. The result for one of the variants [i.e., Chebyshev–Gauss–Lobatto (CGL)] is shown in Fig. 3. Although the CGL result was anticipated, what was surprising about the other variants of the Birkhoff PS method was that it was successful even for Gauss- and Radau-based grid points. In other words, a Birkhoff PS method is apparently able to overcome the well-known deficiencies [7,8,10,26] of Gauss and Radau grids. Its apparent universal efficacy suggests that a significant amount of theoretical analysis remains to be done to understand and advance Birkhoff PS optimal control programming techniques.

VI. Conclusions

The formulation of a pseudospectral (PS) mathematical programming problem for an arbitrary grid is reasonably straightforward; however, a proper implementation of its constituent matrices, grid points, etc., requires a significant amount of sophistication and computational finesse. To advance an efficient implementation of a Birkhoff PS method, many details of the computational process and formulas for a plethora of orthogonal functions and grids have been provided so that various options may be readily tested over sample problems. It is quite critical to understand that, despite its superficial simplicity, a proper implementation of a PS method requires a thorough understanding of the many nuances in optimal control programming. A naive patching of the various computational elements of approximation theory and optimal control can produce poor or even erroneous results. A correct implementation of PS programming techniques can solve a wide variety of singular optimal control problems without requiring a painstaking index-reduction process. To advance a clearer understanding and implementation of the emerging Birkhoff PS methods, a unified grid-agnostic mathematical programming problem formulation has been presented. Preliminary numerical tests indicate a high degree of robustness of the Birkhoff PS method relative to different choices of ultraspherical polynomials and various Gaussian grids. As a means to further stress the apparent robustness of a Birkhoff PS method, a singular optimal control problem was considered as a test case. The particular singular optimal control problem that was chosen in this Note was the Gong problem that has previously been able to discriminate between different discretizations and grid selections. Twelve variants of the Birkhoff PS method produced the correct solution. The computational results suggest that there are likely many deeper aspects of the Birkhoff PS optimal control theory that await discovery.

References

[1] Koeppen, N., Ross, M. I., Wilcox, C. L., and Proulx, J. R., "Fast Mesh Refinement in Pseudospectral Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 4, 2019, pp. 711–722.

[2] Wang, L.-L., Samson, D. M., and Zhao, X., "A Well-Conditioned Collocation Method Using a Pseudospectral Integration Matrix," *SIAM Journal on Scientific Computing*, Vol. 36, No. 3, 2014, pp. A907–A929.

[3] Lorentz, G. G., and Zeller, L. K., "Birkhoff Interpolation," *SIAM Journal on Numerical Analysis*, Vol. 8, No. 1, 1971, pp. 43–48.

[4] Schoenberg, J. I., "On Hermite-Birkhoff Interpolation," *Journal of Mathematical Analysis and Applications*, Vol. 16, No. 3, 1966, pp. 538–543.

[5] Gong, Q., Ross, M. I., and Fahroo, F., "Pseudospectral Optimal Control on Arbitrary Grids," *AAS Astrodynamics Specialist Conference*, AAS Paper 09-405, Springfield, VA, 2009.

[6] Gong, Q., Ross, M. I., and Fahroo, F., "Spectral and Pseudospectral Optimal Control over Arbitrary Grids," *Journal of Optimization Theory and Applications*, Vol. 169, No. 3, 2016, pp. 759–783.

[7] Fahroo, F., and Ross, M. I., "Convergence of the Costates Does Not Imply Convergence of the Control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1492–1497.

[8] Ross, M. I., *A Primer on Pontryagin's Principle in Optimal Control*, 2nd ed., Collegiate Publ., San Francisco, CA, 2015.

[9] Gong, Q., Kang, W., and Ross, M. I., "A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems," *IEEE Transactions on Automatic Control*, Vol. 51, No. 7, July 2006, pp. 1115–1129.

[10] Ross, M. I., and Karpenko, M., “A Review of Pseudospectral Optimal Control: From Theory to Flight,” *Annual Reviews in Control*, Vol. 36, No. 2, 2012, pp. 182–197.

[11] Ross, M. I., Fahroo, F., and Strizzi, J., “Adaptive Grids for Trajectory Optimization by Pseudospectral Methods,” *AAS/AIAA Spaceflight Mechanics Conference*, AAS Paper 03-142, Springfield, VA, Feb. 2003.

[12] Ross, M. I., Gong, Q., Fahroo, F., and Kang, W., “Practical Stabilization Through Real-Time Optimal Control,” *Proceedings of the 2006 American Control Conference*, IEEE Publ., Piscataway, NJ, June 2006, pp. 14–16.

[13] Fahroo, F., and Ross, M. I., “Pseudospectral Methods for Infinite-Horizon Optimal Control Problems,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 927–936.

[14] Ross, M. I., Gong, Q., Karpenko, M., and Proulx, J. R., “Scaling and Balancing for High-Performance Computation of Optimal Controls,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 10, 2018, pp. 2086–2097.

[15] Ross, M. I., Gong, Q., and Sekhavat, P., “Low-Thrust, High-Accuracy Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, 2007, pp. 921–933.

[16] Ross, M. I., Gong, Q., and Sekhavat, P., “The Bellman Pseudospectral Method,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, AIAA Paper 2008-6448, Aug. 2008.

[17] Kosloff, D., and Tal-Ezer, H., “A Modified Chebyshev Pseudospectral Method with an $\mathcal{O}(N^{-1})$ Time Step Restriction,” *Journal of Computational Physics*, Vol. 104, No. 2, 1993, pp. 457–469.

[18] Gong, Q., Fahroo, F., and Ross, M. I., “Spectral Algorithm for Pseudospectral Methods in Optimal Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460–471.

[19] Betts, T. J., *Practical Methods for Optimal Control Using Nonlinear Programming*, Soc. for Industrial and Applied Mathematics, Philadelphia, PA, 2001.

[20] Marsh, C. H., Karpenko, M., and Gong, Q., “A Pseudospectral Approach to High Index DAE Optimal Control Problems,” eprint arXiv:1811.12582, Nov. 2018.

[21] Ross, M. I., “An Optimal Control Theory for Nonlinear Optimization,” *Journal of Computational and Applied Mathematics*, Vol. 354, July 2019, pp. 39–51.

[22] Boyd, J., *Chebyshev and Fourier Spectral Methods*, Dover, Mineola, NY, 2001.

[23] Shen, J., Tang, T., and Wang, L.-L., *Spectral Methods: Algorithms, Analysis and Applications*, Springer, Heidelberg, 2011.

[24] Trefethen, N. L., *Approximation Theory and Approximation Practice*, Soc. for Industrial and Applied Mathematics, Philadelphia, PA, 2013.

[25] Krall, L. H., “On Derivatives of Orthogonal Polynomials,” *Bulletin of the American Mathematical Society*, Vol. 41, No. 6, 1936, pp. 423–428.

[26] Fahroo, F., and Ross, M. I., “Advances in Pseudospectral Methods for Optimal Control,” *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2008-7309, Aug. 2008.