

Method for Solving State-Path Constrained Optimal Control Problems Using Adaptive Radau Collocation

Cale A. Byczkowski*
Anil V. Rao†

*University of Florida
Gainesville, FL 32611-6250*

April 11, 2023

Abstract

A new method is developed for accurately approximating the solution to state-variable inequality path constrained optimal control problems using a multiple-domain adaptive Legendre-Gauss-Radau collocation method. The method consists of the following parts. First, a structure detection method is developed to estimate switch times in the activation and deactivation of state-variable inequality path constraints. Second, using the detected structure, the domain is partitioned into multiple-domains where each domain corresponds to either a constrained or an unconstrained segment. Furthermore, additional decision variables are introduced in the multiple-domain formulation, where these additional decision variables represent the switch times of the detected active state-variable inequality path constraints. Within a constrained domain, the path constraint is differentiated with respect to the independent variable until the control appears explicitly, and this derivative is set to zero along the constrained arc while all preceding derivatives are set to zero at the start of the constrained arc. The time derivatives of the active state-variable inequality path constraints are computed using automatic differentiation and the properties of the chain rule. The method is demonstrated on two problems, the first being a benchmark optimal control problem which has a known analytical solution and the second being a challenging problem from the field of aerospace engineering in which there is no known analytical solution. When compared against previously developed adaptive Legendre-Gauss-Radau methods, the results show that the method developed in this paper is capable of computing accurate solutions to problems whose solution contain active state-variable inequality path constraints.

*Ph.D. Student, Department of Mechanical and Aerospace Engineering. Email: cale.byczkowski@ufl.edu.

†Professor, Department of Mechanical and Aerospace Engineering, University Term Professor. Associate Fellow, AIAA.
E-mail: anilvrao@ufl.edu. Corresponding Author

1 Introduction

An optimal control problem is one where it is desired to optimize a specified performance index while satisfying dynamic constraints, path constraints, and boundary conditions. Since most optimal control problems do not have analytic solutions, numerical methods must be developed and used to approximate the solution of an optimal control problem. Numerical methods for optimal control fall into two broad categories: indirect and direct methods. In an indirect method, the first-order optimality conditions are derived using the calculus of variations and the original optimal control problem is converted to a Hamilton boundary-value problem (HBVP). The HBVP is then solved using either single-shooting, multiple-shooting, or indirect collocation. In a direct method, either the control or the state and control are approximated at a set of support points, and the original optimal control problem is transcribed to a nonlinear programming problem (NLP). The NLP can then be solved using well-developed software such as *SNOPT* [1], *IPOPT* [2], or *KNITRO* [3].

A particular class of direct methods (known as *direct collocation*) has been extensively used for computing the numerical approximation to a continuous time optimal control problem. In a direct collocation method (also referred to as an implicit simulation method), both the state and control are parameterized, and the constraints of the continuous time optimal control problem are enforced at a set of specially chosen *collocation* points. In the past decade, a great deal of research has gone into the development of *Gaussian quadrature orthogonal collocation methods*. In a Gaussian quadrature orthogonal collocation method, the state is approximated using a basis of orthogonal Lagrange polynomials where the specially chosen set of collocation points is associated with a Gaussian quadrature. The most well developed Gaussian quadrature methods employ either Legendre-Gauss (LG) points [4, 5], Legendre-Gauss-Radau (LGR) points [6–9], or Legendre-Gauss-Lobatto (LGL) points [10]. It has been shown that when the solution for the optimal control takes on a smooth form, Gauss quadrature methods are capable of converging to highly accurate solutions at an exponential rate [11]. Conversely, when the solution is nonsmooth or singular, which is often the case when constraints are enforced on the state and/or control, advanced collocation methods tend to require a large amount of computation time and often return poor or no solution for the problem of interest. As a result, standard direct collocation methods often return suboptimal or even unfeasible solutions.

In the past decade, a class of so called *hp*-adaptive mesh refinement methods have been developed that can handle discontinuities in the solution and compute highly accurate solutions [12–16]. Though, it is often the case that an *hp*-adaptive method will place an unnecessary amount of collocation points and mesh intervals around a discontinuity, resulting in a larger amount of computation. A similar behavior is exhibited when a state-variable inequality constraint is present in the problem, thus motivating the development of a method that can properly partition a problem such that the rapid convergence of a Gaussian quadrature collocation method can still be exploited without suffering when a state-variable inequality constraint is present.

Increased-complexity optimal control problems are often subject to path constraints. An important class of path-constrained optimal control problems that poses challenges computationally are problems with *state-variable inequality constraints* (SVICs), henceforth referred to as a state-path constrained optimal control

problem (SPOC). That is, a SPOC is an optimal control problem with inequality constraints that are only a function of the state and possibly time. The added complexity of solving a SPOC typically stems from not knowing when and where the inequality constraints are active on the optimal solution. Given the constraint is only a function of the state (and possibly time), no information on the optimal control is readily available when the constraint is active. Since the early 1960s, a number of researchers have delved into establishing additional necessary conditions for optimality when such constraints are present. This research has led to a variety of different sets of optimality conditions, mainly stemming from three different approaches often referred to as the method of *direct adjoining* [17–20], *indirect adjoining* [21–25], and *indirect adjoining with continuous multipliers* [26–30].

In the direct adjoining approach, the SVICs (in their original form) are adjoined to the Hamiltonian of the system through the use of Lagrange multipliers. Next, the calculus of variations is used to derive first-order optimality conditions which result in a set of "jump conditions" for the optimal costate at the beginning and end of any constrained arcs, leading to possible discontinuities in the costate. As for the indirect adjoining approach, obtaining the necessary conditions for optimality involves adjoining to the Hamiltonian the lowest time derivative of the SVIC which is explicit in the control. Using this approach, a set of necessary "tangency conditions" are enforced at either the entrance, and/or exit of the constrained arc, which also lead to discontinuities in the costate [21]. Lastly, the approach of indirect adjoining with continuous multipliers involves defining a new costate variable so as to "subtract" the discontinuity, leading to a continuous costate [30]. A small class of problems exist for which an analytical solution can be derived when applying the optimality conditions obtained from any of the aforementioned approaches, and for this reason, research for developing numerical methods to approximate the solution to a SPOC has been a focus for decades.

As alluded to earlier, from a numerical point of view, additional challenges exist when trying to approximate the solution to inequality-constrained optimal control problems. For example, when the inequality constraints are purely a function of the control, the optimal control profile may contain discontinuities and singular arcs, which can be difficult to approximate numerically. In the case that inequality constraints are purely a function of the state variables, the problem leads to a system consisting of *high-index* differential-algebraic equation (DAE) constraints when the constraint is active, which are computationally challenging to solve with traditional numerical integration methods [31–33]. Moreover, activations of the SVICs may induce nonsmooth behavior in the state and control solutions while also introducing discontinuities in the costates. This research is motivated by the importance of having a numerical method capable of computing an accurate approximation to problems whose solutions may be nonsmooth due to the existence of SVICs.

Several numerical methods have been developed for solving a SPOC, all similarly falling within the category of either a direct, indirect, or a combination of both; all of which utilize the various forms of the additional necessary conditions for inequality constraints. The most common methods that appear within the indirect approach are so called penalty function techniques [34–36] and transformation techniques [37–39]. In a penalty function method, the original constrained problem is approximated by augmenting the inequality

constraint to the cost function through the use of a penalty term. The approximated problem, which is now unconstrained, is iteratively solved until it approaches the solution of the original problem. In a transformation method, the dimension of the state is increased through the inclusion of parametric variables (also known as “slack variables”). If the order of the inequality constraint is q , then the dimension of the state increases by q , and the additional state variables are defined such that the inequality constraint is satisfied at every point throughout the trajectory. Both penalty term and transcription methods benefit from the fact that no a priori knowledge of when the constraint becomes active is necessary; however, both methods suffer computationally as each need to employ some form of iterative process to obtain an accurate approximate of the solution.

As for handling SVICs in a direct approach, the state variable constraint (or an appropriate time derivative) are directly adjoined to the problem [23, 32, 40, 41]. Specifically, the state path constraint and its $q - 1$ time derivatives are enforced as equality constraints at the entry of the constrained arc, while the q^{th} time derivative is enforced as a mixed state/control equality constraint. A pitfall in using a direct approach stems from the fact that the number and location of constraint activations must be guessed and/or known a priori.

In this paper a new method is presented for solving state-path constrained optimal control problems. The method described in this paper consists of the following parts. First, a structure detection algorithm is developed to detect estimated activation and deactivation times of the SVICs. Second, based on the results of the detection algorithm, the multiple-domain reformulation of the LGR collocation method developed in Ref. [42] is used to partition the problem into subdomains where the detected constraints are either deemed to be active or inactive. Similar to Ref. [42], the multiple-domain formulation partitions the time horizon such that the estimated activation and deactivation times are included as additional decision variables in the optimization process. Third, the method automatically enforces the additional necessary conditions following from the approach of indirect adjoining. Specifically, a method is developed to algorithmically obtain the higher time derivatives of the active SVICs in order to form the necessary tangency conditions which are enforced at the beginning of constrained domains while the highest such time derivative of the active SVICs that is explicit in a control variable is enforced as an equality constraint during the constrained domains. Lastly, the method developed in this paper utilizes adaptive LGR quadrature to suggest refined meshes as well as allow for the ability to re-perform structure detection. This adds an additional level of robustness to the method to allow for the capability to accurately detect the constraint structure.

The contributions of this work are as follows. First, a structure detection algorithm is developed to obtain estimates of activation and deactivation times of any active SVICs. Second, based on the detection results, the method automatically partitions the original time domain into multiple subdomains where the constraints are either set to be active or inactive. Third, the method does not require any a priori knowledge of the structure of the active SVICs. Fourth, the method includes the detected activation and deactivation times within the optimization decision vector, allowing for the exact locations to be optimized. Fifth, the method includes additional necessary conditions for optimality within the direct method formulation. Specifically, the approach of breaking up the original time domain into multiple subdomains allows for the

use of additional necessary conditions to be extended to direct methods.

The remainder of the paper is organized as follows. Section 2 introduces an optimal control problem in Bolza form. Section 3 describes the multiple-domain formulation of Legendre-Gauss-Radau collocation used within this work to formulate the nonlinear programming problem. Section 4 provides an overview of the additional necessary conditions used within this work for handling SVICs. Section 5 details the method developed in this work for solving state-path constrained optimal control problems. Section 6 demonstrates the method on two examples. Section 7 discusses limitations of the method developed in this work. Finally, Section 8 provides conclusions on this work.

2 Bolza Optimal Control Problem

Consider the following general optimal control problem in Bolza form. Determine the state, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$, the control, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, the initial time, t_0 , and the terminal time, t_f , on the time interval, $t \in [t_0, t_f]$, (where n_y and n_u represent the number of states and controls, respectively) that minimizes the cost functional,

$$\mathcal{J} = \mathcal{M}(\mathbf{y}(t_0), t_0, \mathbf{y}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{y}(\mathbf{t}), \mathbf{u}(\mathbf{t}), \mathbf{t}) d\mathbf{t} \quad (1)$$

subject to the dynamic constraints,

$$\frac{d\mathbf{y}}{dt} = \mathbf{a}(\mathbf{y}(t), \mathbf{u}(t), t), \quad (2)$$

the inequality path constraints,

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{y}(t), \mathbf{u}(t), t) \leq \mathbf{c}_{\max} \quad (3)$$

and the boundary conditions,

$$\mathbf{b}_{\min} \leq \mathbf{b}(\mathbf{y}(t_0), t_0, \mathbf{y}(t_f), t_f) \leq \mathbf{b}_{\max} \quad (4)$$

The functions \mathcal{M} , \mathcal{L} , \mathbf{a} , \mathbf{b} , \mathbf{c} are defined by the following mappings,

$$\begin{aligned} \mathcal{M} &: \mathbb{R}^{n_y} \times \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R} \rightarrow \mathbb{R}, \\ \mathcal{L} &: \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}, \\ \mathbf{a} &: \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}^{n_y}, \\ \mathbf{b} &: \mathbb{R}^{n_y} \times \mathbb{R} \times \mathbb{R}^{n_y} \times \mathbb{R} \rightarrow \mathbb{R}^{n_b}, \\ \mathbf{c} &: \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}^{n_c}, \end{aligned}$$

where n_b and n_c are the number of specified boundary conditions and inequality path constraints, respectively. From a numerical standpoint, and without loss of generality, it is useful to map the original time interval, $t \in [t_0, t_f]$, of the Bolza optimal control problem onto the time interval $\tau \in [-1, +1]$. This is done with the use of the following affine transformation,

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \quad (5)$$

The Bolza optimal control problem of Eqs.(1-4) can now be defined in terms of τ as follows. Determine the state, $\mathbf{y}(\tau) \in \mathbb{R}^{n_y}$, the control, $\mathbf{u}(\tau) \in \mathbb{R}^{n_u}$, the initial time, t_0 , and the terminal time, t_f , on the time

interval, $\tau \in [-1, +1]$, that minimizes the cost functional,

$$\mathcal{J} = \mathcal{M}(\mathbf{y}(-1), t_0, \mathbf{y}(+1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{+1} \mathcal{L}(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) d\tau \quad (6)$$

subject to the dynamic constraints,

$$\frac{d\mathbf{y}}{d\tau} = \frac{t_f - t_0}{2} \mathbf{a}(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \quad (7)$$

the inequality path constraints,

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \leq \mathbf{c}_{\max} \quad (8)$$

and the boundary conditions,

$$\mathbf{b}_{\min} \leq \mathbf{b}(\mathbf{y}(-1), t_0, \mathbf{y}(+1), t_f) \leq \mathbf{b}_{\max} \quad (9)$$

Suppose now that the interval $\tau \in [-1, +1]$ is divided into K mesh intervals $\mathcal{S}_k = [\tau_{k-1}, \tau_k]$, $k = 1, \dots, K$, where (τ_0, \dots, τ_K) are the mesh points. The mesh points have the property that $-1 = \tau_0 < \tau_1 < \tau_2 < \dots < \tau_{K-1} < \tau_K = +1$. Next, let $\mathbf{y}^{(k)}(\tau)$ and $\mathbf{u}^{(k)}(\tau)$ be the state and control in the mesh interval \mathcal{S}_k . Then the Bolza optimal control problem of Eqs.(6-9) can be rewritten as follows. Determine the state, $\mathbf{y}^{(k)}(\tau) \in \mathbb{R}^{n_y}$, the control, $\mathbf{u}^{(k)}(\tau) \in \mathbb{R}^{n_u}$, the initial time t_0 , and the final time, t_f , on the interval $\tau \in [-1, +1]$ for $k = 1, \dots, K$ that minimizes the cost functional,

$$\mathcal{J} = \mathcal{M}(\mathbf{y}^{(1)}(-1), t_0, \mathbf{y}^{(K)}(+1), t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^K \int_{\tau_{k-1}}^{\tau_k} \mathcal{L}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_0, t_f) d\tau, \quad (k = 1, \dots, K) \quad (10)$$

subject to the dynamic constraints,

$$\frac{d\mathbf{y}^{(k)}(\tau)}{d\tau} = \frac{t_f - t_0}{2} \mathbf{a}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_0, t_f), \quad (k = 1, \dots, K) \quad (11)$$

the inequality path constraints,

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_0, t_f) \leq \mathbf{c}_{\max}, \quad (k = 1, \dots, K) \quad (12)$$

and the boundary conditions,

$$\mathbf{b}_{\min} \leq \mathbf{b}(\mathbf{y}^{(1)}(-1), t_0, \mathbf{y}^{(K)}(+1), t_f) \leq \mathbf{b}_{\max}, \quad (k = 1, \dots, K) \quad (13)$$

Since it is necessary that the state be continuous at the interior mesh points, the condition $\mathbf{y}(\tau_k^-) = \mathbf{y}(\tau_k^+)$ holds for $k = (1, \dots, K - 1)$.

3 Multiple-Domain Legendre-Gauss-Radau Collocation

The method developed in this paper utilizes Legendre-Gauss-Radau (LGR) collocation to discretize the continuous Bolza optimal control problem. The LGR collocation method is chosen due to its ability to converge to a solution at an exponential rate when the solution is smooth [11]. However, for the problems

studied in this research, the solution may be nonsmooth due to active state-constrained arcs. Consequently, in order to be able to correctly enforce additional necessary conditions and handle potential nonsmooth behavior, the method developed in this research utilizes a multiple-domain LGR collocation method (similar to Pager, et. al. [42]) described below.

3.1 Multiple-Domain Formulation

Let the horizon $t \in [t_0, t_f]$ of the Bolza optimal control problem be partitioned into D domains, $\mathcal{P}_d = [t_s^{\{d-1\}}, t_s^{\{s\}}] \subseteq [t_0, t_f]$, $d \in [1, \dots, D]$, such that,

$$\begin{aligned}\bigcup_{d=1}^D \mathcal{P}_d &= [t_0, t_f], \\ \bigcap_{d=1}^D \mathcal{P}_d &= [t_s^{\{1\}}, \dots, t_s^{\{D-1\}}],\end{aligned}\tag{14}$$

where $t_s^{\{d\}}$, are the *domain interface variables*, $t_s^{\{0\}} = t_0$, $t_s^{\{D\}} = t_f$, are the initial and final times, respectively, d is the domain index, and s signifies a switch between a constrained or unconstrained arc. Specifically, for the method developed in this research, $t_s^{\{d\}}$ for $d \neq \{0, D\}$ correspond to switches in activation and deactivation of the state-variable inequality constraint, whereas in Pager, et. al. [42], these variables serve as switch points in bang-bang and/or singular control structures. It is noted that the domain interface variables become additional decision variables in the resulting nonlinear programming problem (NLP) and are *not* collocation points.

Similar to the formulation presented before, each domain $\mathcal{P}_d = [t_s^{\{d-1\}}, t_s^{\{s\}}]$ is mapped to $\tau \in [-1, +1]$ using the affine transformation,

$$t = \frac{t_s^{\{d\}} - t_s^{\{d-1\}}}{2}\tau + \frac{t_s^{\{d\}} + t_s^{\{d-1\}}}{2}\tag{15}$$

Next, the time interval $\tau \in [-1, +1]$ on each domain is divided into K mesh intervals, $\mathcal{S}_k = [T_{k-1}, T_k] \subseteq [-1, +1]$, for $k = [1, \dots, K]$, such that,

$$\begin{aligned}\bigcup_{k=1}^K \mathcal{S}_k &= [\tau_0, \tau_f], \\ \bigcap_{k=1}^K \mathcal{S}_k &= [T_1, \dots, T_{K-1}].\end{aligned}\tag{16}$$

3.2 Legendre-Gauss-Radau Collocation

Once the multiple-domain formulation is complete, LGR collocation is used to discretize the continuous time problem on each mesh interval. Specifically, for each mesh interval, the LGR points are defined on $[T_{k-1}, T_k] \subseteq [-1, +1]$, $k = [1, \dots, K]$. The state of the continuous time optimal control problem is then approximated within each mesh interval \mathcal{S}_k by,

$$\mathbf{y}^{(k)}(\tau) \approx \mathbf{Y}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{Y}_j^{(k)} \ell_j^{(k)}(\tau),\tag{17}$$

where $\ell_j^{(k)}(\tau)$, $j = [1, \dots, N_k + 1]$, is a basis of Lagrange polynomials on \mathcal{S}_k given by,

$$\ell_j^{(k)}(\tau) = \prod_{\substack{i=1 \\ i \neq j}}^{N_k+1} \frac{\tau - \tau_i^{(k)}}{\tau_j^{(k)} - \tau_i^{(k)}},\tag{18}$$

where $(\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)})$ are the N_k set of LGR collocation points in the interval $[T_{k-1}, T_k)$, with $\tau_{N_k+1} = T_k$ being a non-located support point. Differentiating Eq. (17) with respect to τ leads to,

$$\frac{d\mathbf{Y}^{(k)}(\tau)}{d\tau} = \sum_{j=1}^{N_k+1} \mathbf{Y}_j^{(k)} \frac{d\ell_j^{(k)}(\tau)}{d\tau}. \quad (19)$$

The dynamic constraints given in Eq. (7) are then approximated at the N_k -LGR points in mesh interval \mathcal{S}_k , $k = [1, \dots, K]$ on domain $d = [1, \dots, D]$ by,

$$\sum_{j=1}^{N_k+1} D_{ij}^{(k)} \mathbf{Y}_j^{(k)} - \frac{t_f - t_0}{2} \mathbf{a}(\mathbf{Y}_i^{(k)}, \mathbf{U}_i^{(k)}, t(\tau_i^{(k)}, t_s^{\{d-1\}}, t_s^{\{d\}})) = 0, \quad \forall i \in \{1, \dots, N_k\} \quad (20)$$

where,

$$D_{ij}^{(k)} = \frac{d\ell_j^{(k)}(\tau)}{d\tau}, \quad \forall i \in \{1, \dots, N_k\}, \forall j \in \{1, \dots, N_k + 1\} \quad (21)$$

are the elements of the $N_k \times (N_k + 1)$ Legendre-Gauss-Radau *differentiation matrix* in the mesh interval \mathcal{S}_k , and $\mathbf{U}_i^{(k)}$ is the approximation of the control at the i -th collocation point in mesh interval \mathcal{S}_k . Continuity in the state across mesh intervals $\mathcal{S}_{k-1}, \mathcal{S}_k$ and time domains $\mathcal{P}_{d-1}, \mathcal{P}_d$ is achieved by using the same variable to represent $\mathbf{Y}_{N_{k-1}+1}^{k-1} = \mathbf{Y}_1^{(k)}$ and $\mathbf{Y}_{N^{\{d-1\}}+1}^{\{d-1\}} = \mathbf{Y}_1^{\{d\}}$, respectively. Lastly, it is noted that $N^{\{d\}}$ is the total number of collocation points in the time domain \mathcal{P}_d and can be computed by,

$$N^{\{d\}} = \sum_{k=1}^{K^{\{d\}}} N_k^{\{d\}} \quad (22)$$

where $K^{\{d\}}$ is the total number of mesh intervals in time domain \mathcal{P}_d , $d = \{1, \dots, D\}$.

3.3 Nonlinear Programming Problem

The multiple-domain Legendre-Gauss-Radau (LGR) discretization of the continuous time multiple-domain Bolza optimal control problem results in a large sparse nonlinear programming (NLP) which is given as follows. Minimize the objective functional,

$$\mathcal{J} = \mathcal{M}(\mathbf{Y}_1^{\{1\}}, t_0, \mathbf{Y}_{N^{\{D\}}+1}^{\{D\}}, t_f) + \sum_{d=1}^D \frac{t_s^{\{d\}} - t_s^{\{d-1\}}}{2} [\mathbf{w}^{\{d\}}]^\top \mathbf{L}^{\{d\}} \quad (23)$$

subject to the dynamic constraints,

$$\Delta^{\{d\}} = \mathbf{D}^{\{d\}} \mathbf{Y}^{\{d\}} - \frac{t_s^{\{d\}} - t_s^{\{d-1\}}}{2} \mathbf{A}^{\{d\}} = \mathbf{0} \quad \forall d \in \{1, \dots, D\} \quad (24)$$

the inequality path constraints,

$$\mathbf{c}_{\min} \leq \mathbf{C}^{\{d\}} \leq \mathbf{c}_{\max}, \quad \forall d \in \{1, \dots, D\} \quad (25)$$

the boundary conditions,

$$\mathbf{b}_{\min} \leq \mathbf{b}(\mathbf{Y}_1^{\{1\}}, t_0, \mathbf{Y}_{N^{\{d\}}+1}^{\{d\}}, t_f) \leq \mathbf{b}_{\max} \quad (26)$$

and the continuity conditions,

$$\mathbf{Y}_{N^{\{d-1\}}+1}^{\{d-1\}} = \mathbf{Y}_1^{\{d\}} \quad (27)$$

where $\mathbf{D}^{\{d\}} \in \mathbb{R}^{N^{\{d\}} \times (N^{\{d\}} + 1)}$ is the LGR differentiation matrix in the time domain \mathcal{P}_d , $d = [1, \dots, D]$, and $\mathbf{w}^{\{d\}} \in \mathbb{R}^{N^{\{d\}} \times 1}$ are the LGR weights at each collocation point in the time domain \mathcal{P}_d . Note, the continuity conditions in Eq. (27) are implicitly enforced by using the same decision variable in the NLP for $\mathbf{Y}_{N^{\{d-1\}}+1}^{\{d-1\}}$ and $\mathbf{Y}_1^{\{d\}}$. The elements, $\mathbf{A}^{\{d\}}, \mathbf{C}^{\{d\}}, \mathbf{L}^{\{d\}}$, in Eqs. (23)-(25) are defined as follows,

$$\begin{aligned} \mathbf{A}^{\{d\}} &= \begin{bmatrix} \mathbf{a}(\mathbf{Y}_1^{\{d\}}, \mathbf{U}_1^{\{d\}}, t_1^{\{d\}}) \\ \vdots \\ \mathbf{a}(\mathbf{Y}_{N^{\{d\}}}^{\{d\}}, \mathbf{U}_{N^{\{d\}}}^{\{d\}}, t_{N^{\{d\}}}^{\{d\}}) \end{bmatrix} \in \mathbb{R}^{N^{\{d\}} \times n_x}, \\ \mathbf{C}^{\{d\}} &= \begin{bmatrix} \mathbf{c}(\mathbf{Y}_1^{\{d\}}, \mathbf{U}_1^{\{d\}}, t_1^{\{d\}}) \\ \vdots \\ \mathbf{c}(\mathbf{Y}_{N^{\{d\}}}^{\{d\}}, \mathbf{U}_{N^{\{d\}}}^{\{d\}}, t_{N^{\{d\}}}^{\{d\}}) \end{bmatrix} \in \mathbb{R}^{N^{\{d\}} \times n_c}, \\ \mathbf{L}^{\{d\}} &= \begin{bmatrix} \mathcal{L}(\mathbf{Y}_1^{\{d\}}, \mathbf{U}_1^{\{d\}}, t_1^{\{d\}}) \\ \vdots \\ \mathcal{L}(\mathbf{Y}_{N^{\{d\}}}^{\{d\}}, \mathbf{U}_{N^{\{d\}}}^{\{d\}}, t_{N^{\{d\}}}^{\{d\}}) \end{bmatrix} \in \mathbb{R}^{N^{\{d\}} \times 1}, \end{aligned} \quad (28)$$

Additionally, the state matrix, $\mathbf{Y}^{\{d\}} \in \mathbb{R}^{(N^{\{d\}} + 1) \times n_x}$, and the control matrix, $\mathbf{U}^{\{d\}} \in \mathbb{R}^{N^{\{d\}} \times n_u}$, in time domain \mathcal{P}_d , $d \in \{1, \dots, D\}$, are formed as

$$\mathbf{Y}^{\{d\}} = \begin{bmatrix} \mathbf{Y}_1^{\{d\}} \\ \vdots \\ \mathbf{Y}_{N^{\{d\}}+1}^{\{d\}} \end{bmatrix} \quad \text{and} \quad \mathbf{U}^{\{d\}} = \begin{bmatrix} \mathbf{U}_1^{\{d\}} \\ \vdots \\ \mathbf{U}_{N^{\{d\}}}^{\{d\}} \end{bmatrix} \quad (29)$$

respectively, where n_x is the number of state components and n_u is the number of control components in the problem.

3.4 Approximation of Solution Error

Suppose the nonlinear program of Eqs. (23)-(27) has been solved on a mesh, \mathcal{S}_k , $k = 1, \dots, K$, on domain \mathcal{P}_d , $d = 1, \dots, D$, with N_k collocation points in mesh interval \mathcal{S}_k . An estimate of the discretization error on the current mesh must be obtained in order to assess the accuracy of the solution. The method developed in this paper directly uses the maximum estimated relative error in the state solution as a condition for terminating the algorithm. The process used for estimating the relative error is identical to Ref. [14]. The process is summarized as follows.

The objective is to approximate the error in the state at a set of $M_k = N_k + 1$ LGR points $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_{M_k}^{(k)})$, where $\hat{\tau}_1^{(k)} = \tau_1^{(k)} = T_{k-1}$ and $\hat{\tau}_{M_k+1}^{(k)} = T_k$. Let the state approximation at the M_k LGR points be denoted by $(\mathbf{Y}(\hat{\tau}_1^{(k)}), \dots, \mathbf{Y}(\hat{\tau}_{M_k}^{(k)}))$. Next, let the control be approximated in mesh interval \mathcal{S}_k with a Lagrange polynomial,

$$\mathbf{U}^{(k)}(\tau) = \sum_{j=1}^{N_k} \mathbf{U}_j^{(k)} \hat{\ell}_j^{(k)}(\tau), \quad \hat{\ell}_j^{(k)}(\tau) = \prod_{\substack{l=1 \\ l \neq j}}^{N_k} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}}, \quad (30)$$

and let the control approximation at $\hat{\tau}_i^{(k)}$ be denoted $\mathbf{U}(\hat{\tau}_i^{(k)})$ for $i = 1, \dots, M_k$, then the value of the right-hand side of the dynamics at $(\mathbf{Y}(\hat{\tau}_i^{(k)}), \mathbf{U}(\hat{\tau}_i^{(k)}), \hat{\tau}_i^{(k)})$ is used to construct an improved approximation of the state. Let $\hat{\mathbf{Y}}^{(k)}$ be a polynomial of degree at most M_k that is defined on the mesh interval \mathcal{S}_k . If the derivative of $\hat{\mathbf{Y}}^{(k)}$ matches the dynamics at each LGR quadrature points $\hat{\tau}_i^{(k)}$, $i = 1, \dots, M_k$, then we have,

$$\hat{\mathbf{Y}}^{(k)}(\hat{\tau}_{j+1}^{(k)}) = \mathbf{Y}^{(k)}(\hat{\tau}_1^{(k)}) + \frac{t_f - t_0}{2} \sum_{l=1}^{M_k} \hat{I}_{jl}^{(k)} \mathbf{a}(\mathbf{Y}^{(k)}(\hat{\tau}_l^{(k)}), \mathbf{U}^{(k)}(\hat{\tau}_l^{(k)}), t(\hat{\tau}_l^{(k)}, t_0, t_f)), \quad j = 1, \dots, M_k, \quad (31)$$

where $\hat{I}_{jl}^{(k)}$, $j, l = 1, \dots, M_k$, is the $M_k \times M_k$ LGR integration matrix corresponding to the LGR points defined by $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_{M_k}^{(k)})$. Comparing the interpolated values, $\mathbf{Y}(\hat{\tau}_l^{(k)})$, $l = 1, \dots, M_k + 1$, with the integrated values $\hat{\mathbf{Y}}(\hat{\tau}_l^{(k)})$, $l = 1, \dots, M_k + 1$, the *relative error* in the i^{th} component of the state at $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_{M_k+1}^{(k)})$ is defined as

$$e_i^{(k)}(\hat{\tau}_l^{(k)}) = \frac{|\hat{Y}_i^{(k)}(\hat{\tau}_l^{(k)}) - Y_i^{(k)}(\hat{\tau}_l^{(k)})|}{1 + \max_{\substack{j \in [1, \dots, N_k+1] \\ k \in [1, \dots, K]}} |Y_i^{(k)}(\tau_j^{(k)})|}, \quad \left\{ \begin{array}{l} l = 1, \dots, M_k + 1 \\ i = 1, \dots, n_y \end{array} \right\}. \quad (32)$$

The *maximum relative error* on domain \mathcal{P}_d in mesh interval \mathcal{S}_k is then defined as

$$e_{\max}^{(k)} = \max_{\substack{i \in [1, \dots, n_y] \\ l \in [1, \dots, M_k+1]}} e_i^{(k)}(\hat{\tau}_l^{(k)}) \quad (33)$$

4 State-Variable Inequality Constrained Optimal Control

Suppose now that the inequality path constraint given by Eq. (3) takes on the following form,

$$\mathbf{c}(\mathbf{y}(t), t) \leq \mathbf{0}. \quad (34)$$

In order to determine a condition for the optimal control when the inequality in Eq. (34) holds, successive time derivatives of Eq. (34) are taken, and the dynamic constraints $\mathbf{a}(\mathbf{y}(t), \mathbf{u}(t), t)$ are substituted for $d\mathbf{y}(t)/dt$ until an expression explicit in $\mathbf{u}(t)$ is obtained. This approach is commonly referred to as the method of indirect adjoining [21–25]. If q successive time derivatives are required, then the state-variable inequality path constraints (SVICs) given by Eq. (34) are said to be of q th-order. The Hamiltonian for the system becomes

$$\mathcal{H} = \mathcal{L} + \boldsymbol{\lambda}^T \mathbf{a} + \boldsymbol{\mu}^T \mathbf{c}^{(q)}, \quad (35)$$

where $\boldsymbol{\lambda}$ are the costate variables, $\boldsymbol{\mu}$ are the Lagrange multipliers associated with the SVICs that satisfy the following necessary conditions,

$$\boldsymbol{\mu} \geq \mathbf{0}, \text{ on the constraint boundary } (\mathbf{c}^{(q)} = \mathbf{0}), \quad (36)$$

$$\boldsymbol{\mu} = \mathbf{0}, \text{ off the constraint boundary } (\mathbf{c} < \mathbf{0}), \quad (37)$$

and

$$\mathbf{c}^{(q)} = \frac{d^{(q)} \mathbf{c}}{dt^{(q)}}. \quad (38)$$

The first-order optimality conditions then become

$$\dot{\lambda}^\top = -\frac{\partial \mathcal{H}}{\partial \mathbf{y}} = \begin{cases} -\frac{\partial \mathcal{L}}{\partial \mathbf{y}} - \lambda^\top \frac{\partial \mathbf{a}}{\partial \mathbf{y}} - \mu \frac{\partial \mathbf{c}^{(q)}}{\partial \mathbf{y}}, & \mathbf{c} = \mathbf{0} \\ -\frac{\partial \mathcal{L}}{\partial \mathbf{y}} - \lambda^\top \frac{\partial \mathbf{a}}{\partial \mathbf{y}}, & \mathbf{c} < \mathbf{0} \end{cases} \quad (39)$$

In order to compute a finite control to keep the system on the constraint boundary, the path entering the constraint must satisfy the following *tangency* constraints [21]

$$\mathbf{N}(\mathbf{y}(t_a), t_a) \equiv \begin{bmatrix} \mathbf{c}(\mathbf{y}, t_a) \\ \mathbf{c}^{(1)}(\mathbf{y}, t_a) \\ \vdots \\ \mathbf{c}^{(q-1)}(\mathbf{y}, t_a) \end{bmatrix} = \mathbf{0}, \quad (40)$$

where t_a is the time at which the SVICs become active. It has been shown in [31–33] that when a constraint is active, the index of the original system of differential algebraic equations (DAEs) increases. In a direct method, enforcing Eq. (40) at the start *or* end of a constrained arc is analogous to performing index reduction. Specifically, the tangency constraints provide a set of necessary conditions that reduce the index of the constrained system of DAEs to the index of the unconstrained system. Although direct methods generally perform well when path constraints are present in the problem, it is not possible to enforce the tangency conditions at the correct instances throughout the trajectory without prior knowledge of when the SVICs become active or inactive. The challenge of being able to apply the tangency constraints within a direct method motivates the development of the method presented in this work, which is described in the next section.

5 Method for State-Path Constrained Optimal Control Problems

In this section the method for solving state-path constrained optimal control problems (SPOC) is presented, and will be called the SPOC method. Recall, an optimal control problem with inequality path constraints only a function of the state and possibly time will be referred henceforth as a SPOC. The SPOC method is comprised of two stages. The first stage, described in further detail in Section 5.1, employs a structure detection algorithm to identify activation and deactivation (A/D) times in each state-variable inequality path constraint (SVIC). Next, the detected A/D times are treated as domain interface variables and are used to partition the domain of the independent variable into subdomains such that each subdomain consists of segments where the detected constraint is either active or inactive. The domain interface variables are then treated as additional decision variables in the nonlinear programming problem (NLP) in order to optimize the location(s) for when the detected constraint is active. The next stage of the method described in Section 5.2 enforces additional necessary conditions for optimality such that the partitioned problem is properly constrained. Specifically, time derivatives of the detected constraints are used to reduced the order of the

differential algebraic system of equations and properly constrain the allowable control search space. The final step in the method, as shown in Section 5.3, is to determine if it is necessary to perform mesh refinement and/or structure detection and decomposition to improve the accuracy of the solution.

5.1 Structure Detection and Decomposition

The discussion to follow will assume a single SVIC is present in the problem formulation, but the method developed in this research can be generalized to handle multiple SVICs. Assume now that the optimal control problem formulated in Sect. 2 subject to a SVIC of the form defined in Sect. 4 has been transcribed into a nonlinear programming problem (NLP) using the multiple-domain LGR collocation method provided in Sect. 3 with $D = 1$ (that is, a single domain is used). The solution of the NLP results in a discrete approximation of the state and control, as given by Eq. (29). Assume further that the maximum relative error does not meet the prescribed mesh refinement accuracy tolerance. As a result, mesh refinement is required which simultaneously enables the decomposition of the independent variable into subdomains where the SVIC is either active or inactive. The subdomains are formulated using structure detection as described below.

Structure detection obtains estimates of activation and deactivation (A/D) times in the SVIC on a provided mesh (in this case, the initial mesh) in order to classify subdomains as either constrained or unconstrained. In this work, only the approximation of the SVIC (given by Eq. (28)) is used in the detection process. Structure detection begins by applying the method of Sect. 5.1.1 to identify and estimate any A/D times in the SVIC. After A/D locations have been estimated, the method of Sect. 5.1.2 takes the estimated A/D times and determines the classification of each domain as either constrained or unconstrained. The structure detection process developed in this section is always applied on the initial mesh. Structure detection is only applied recursively if the maximum constraint violations on the current mesh are larger than a user specified constraint satisfaction tolerance (ϵ_c). It is important to note that structure detection is the crux of the method developed in this work and the ability to recursively perform structure detection will ensure an accurate estimate of the true A/D times can be obtained.

5.1.1 Structure Detection

The overall approach used in this research aims to automatically decompose the original inequality constrained optimal control problem into a multiple-domain optimal control problem consisting of state constrained and unconstrained domains. Partitioning the problem into multiple domains provides the ability to apply the higher order necessary tangency conditions at the beginning of constrained domains. The multiple-domain formulation also allows the resulting mixed state-control equality constraint (lowest time derivative of the path constraint that first contains the control) to be enforced throughout constrained domains. Applying the higher order necessary conditions reduces the high-index system of differential algebraic equations (DAEs), and provides the ability to accurately approximate the control while any SVICs are active.

The first step in partitioning the problem is to obtain an estimate of the domain interface variables, which are analogous to activation/deactivation (A/D) times of any SVICs. Before presenting the structure

detection method for estimating the domain interface variables, the following assumption is made in the development of the algorithm:

Assumption 1: The SVICs are not active at the start of the trajectory: $\mathbf{c}(\mathbf{y}(t_0), t_0) < \mathbf{0}$.

The above assumption is in place to be able to inspect neighboring collocation points to determine the activation of a constraint arc and properly enforce the necessary tangency conditions at the start of a constrained domain. It is noted that in Jacobson, et. al. [17], a set of new necessary conditions were derived which show that, in certain cases, for SVICs of *odd* order ($q > 1$), the state may *not* be constrained over a non-zero interval of time, but instead only touch the constraint boundary at a single point. This occurrence is often referred to as a *touch point*. It is mentioned here that the structure detection method developed in this research is intended for SVICs that contain constrained arcs over non-zero intervals of time. That is, the potential existence of touch points are not studied in this research. The SPOC method begins by computing an approximate solution on a single domain $\mathcal{P}_1 = [t_0, t_f]$ using standard Legendre-Gauss-Radau (LGR) collocation on a fixed mesh consisting of $S_k, k = [1, \dots, K]$ mesh intervals with N_k number of collocation points within each mesh interval. Next, let $\mathbf{Y}(\tau_j^{(k)}), j \in \{1, \dots, N_k + 1\}, k \in \{1, \dots, K\}$, be the approximated state solution obtained on the initial domain at the j^{th} collocation point in the k^{th} mesh interval, then the following relative difference is computed at every collocation point plus the final non-collocated point for every SVIC in the problem

$$\delta c_i(\tau_j^{(k)}) = \frac{|c_i(\mathbf{Y}(\tau_j^{(k)}), \tau_j^{(k)}) - c_{i,\max}|}{1 + |c_{i,\max}|}, \quad \forall j \in \{1, \dots, N_k + 1\}, \forall k \in \{1, \dots, K\}, \quad (41)$$

where c_i is the i -th SVIC for all $i = [1, \dots, n_c]$ with n_c being the total number of SVICs. Once the relative difference in Eq. (41) is obtained at every collocation point plus the final non-collocated point, the structure detection algorithm for estimating the activation/deactivation points is executed as follows:

Detection Algorithm for State-Inequality Path Constrained Optimal Control Problems

Step 1: Set detection tolerance for which renders a constraint active: ϵ .

Step 2: Check relative difference to identify activation/deactivation points: $\forall j \neq \{1\}, \forall k \neq \{1\}$

(a): If $\delta c_i(\tau_j^{(k)}) \leq \epsilon$, and

(i): $\delta c_i(\tau_{j-1}^{(k)}) > \epsilon$, and $\delta c_i(\tau_{j+1}^{(k)}) \leq \epsilon$, then $\tau_j^{(k)} = \tau_s$ is deemed an *activation* point.

(ii): $\delta c_i(\tau_{j-1}^{(k)}) \leq \epsilon$, and $\delta c_i(\tau_{j+1}^{(k)}) > \epsilon$, then $\tau_j^{(k)} = \tau_s$ is deemed a *deactivation* point.

Step 3: Compute bounds on estimated activation/deactivation points: $(\tau_s)^\pm$

The default detection tolerance that is used in the detection algorithm is chosen such that it follows a process analogous to selecting the NLP solver tolerance. Specifically, the NLP tolerance is typically chosen to

be on the order of the square root of machine tolerance ($\sim \mathcal{O}(10^{-8})$), and so the default detection tolerance is chosen similarly to be on the order of the square root of the NLP solver tolerance ($\sim \mathcal{O}(10^{-4})$) given the solution on the initial mesh is typically governed by the accuracy at which the NLP solver can achieve. Lastly, it is noted that the final step in the detection algorithm is performed because the detected A/D times are included as additional decision variables in the resulting nonlinear programming problem (NLP) and must be bounded to prevent overlapping or collapsing of domains within the partitioned problem. The bounds are obtained by

$$\left. \begin{aligned} \tau_s^- &= \tau_j^{(k)} + \nu \left(\tau_{j-1}^{(k)} - \tau_j^{(k)} \right), & s &= \{1, \dots, S\}, \\ \tau_s^+ &= \tau_j^{(k)} + \nu \left(\tau_{j+1}^{(k)} - \tau_j^{(k)} \right), & j &= \{1, \dots, N_k\}, \\ & & k &= \{1, \dots, K\}. \end{aligned} \right\} \quad (42)$$

where $S = D - 1$ is the number of detected A/D times and $\nu > 0$ is a user defined parameter used to control the size of the allowable search space for the NLP solver to determine the optimal A/D times. The final step is to use the affine transformation given by Eq. (15) to map the estimated A/D times back onto the original time domain $t \in [t_0, t_f]$.

5.1.2 Structure Decomposition

Assuming the structure detection method of Sect. 5.1.1 has identified activation and deactivation (A/D) times in the SVICs, the next step is to decompose the initial mesh into the multiple-domain structure. Specifically, the detected A/D times directly correlate to the number of domain interface variables, $t_s^{\{d\}}$, $d = \{1, \dots, D-1\}$, to be solved for on subsequent mesh iterations, where the initial guess for each variable is the estimated A/D times τ_s , $s = \{1, \dots, S\}$ obtained from the method of Sect. 5.1.1. The domain interface variables are included as additional decision variables in the NLP decision vector. Recall, the domain interface variables define the start of new domains, $\mathcal{P}^{\{d\}} = [t_s^{\{d\}}, t_s^{\{d-1\}}]$, $d = \{1, \dots, D-1\}$, and are used to partition the time horizon $t \in [t_0, t_f]$ of the original optimal control problem into D domains as described in Sect. 3.

The next step is to enforce an additional constraint on the domain interface variables by including lower and upper bounds to prevent collapsing or overlapping of subdomains. The lower and upper bounds on each interval are determined by taking the bounds given by Eq. (42) and transforming them to the time interval $t \in [t_0, t_f]$ using the affine transformation of Eq. (15). Thus, the bounds for each domain interface variable $[\tau_s^-, \tau_s^+]$, $s = \{1, \dots, S\}$ are transformed to $[t_l^{d-1}, t_u^d]$, $d = \{1, \dots, D-1\}$.

The overall approach for partitioning the time horizon of the original optimal control problem into multiple subdomains is shown in Fig. 1. Although the domain interface variables are used to classify each subdomain, their precise locations are not fixed. The process of including the switch times as additional decision variables in the resulting NLP decision vector allows for their precise locations to be optimized. Recall, the method developed in this research assumes the SVICs are active over a non-zero interval of time, meaning no touch points exist. Thus, the SVIC will be active over a defined interval of time which will produce *at least* one subdomain where the constraint(s) is active (as depicted in Fig. 1). It is noted here that if the maximum relative error computed on the initial mesh ($D = 1$) does not meet the specified mesh

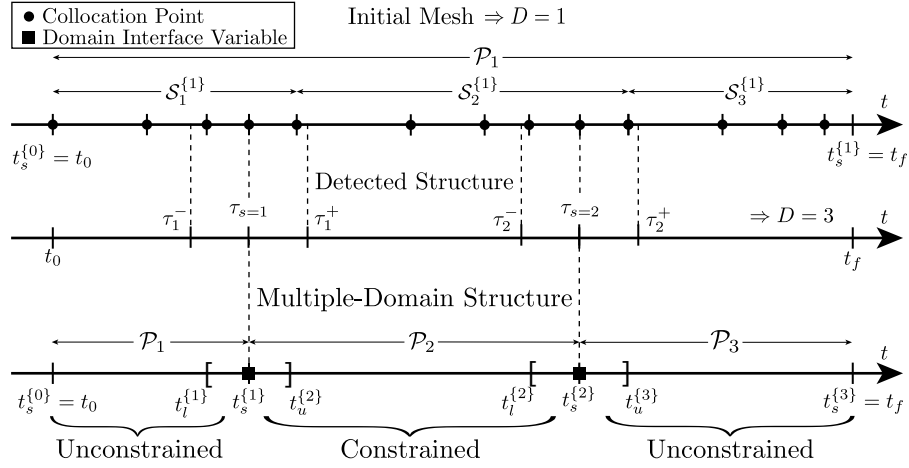


Figure 1: Schematic of process for decomposing the original optimal control problem into D domains where the $D-1$ domain interface variables are included as optimization variables to determine the optimal activation and deactivation in each SVIC.

tolerance, a mesh refinement method is used to obtain a suggested refined mesh. The provided refined mesh is then used when defining the mesh intervals on the partitioned problem.

5.2 Domain Constraints and Refinement

Now that the time domain of the previous mesh has been partitioned into subdomains by the structure detection and decomposition method of Sect. 5.1, the next step is to enforce additional necessary conditions in the subdomains where SVICs have been identified to be active. Once the additional necessary conditions described in Sect. 5.2.1 are properly enforced, the resulting nonlinear programming problem (NLP) is solved. Lastly, the maximum constraint violation and the maximum relative error is analyzed to determine if the domain refinement methods of Sect. 5.2.3 or Sect. 5.2.2 must be employed, respectively.

5.2.1 Additional Necessary Conditions

As described in Sect. 4, the method developed in this work employs the method of indirect adjoining for handling the presence of state-variable inequality path constraints (SVICs). The method of indirect adjoining requires that the q -th time derivative of the SVIC be adjoined to the augmented Hamiltonian of the system (as shown in Eq. (35)). In a direct method for discretizing an optimal control problem, the method of indirect adjoining is equivalent to enforcing the q -th time derivative of an active SVIC as an equality constraint throughout the constrained domain

$$\frac{d^{(q)}c}{dt^{(q)}} = c^{(q)}(\mathbf{y}(\bar{t}), \mathbf{u}(\bar{t}), \bar{t}) = 0, \quad \bar{t} \in [t_a, t_d] \quad (43)$$

where t_a and t_d are the activation and deactivation times of the SVIC, respectively. Additionally, given that the q -th time derivative of the SVIC is being used to derive the optimal control, the path entering

the constrained arc must satisfy the necessary tangency constraints (given by Eq. (40)) [21]. It is assumed that the functions defining the SVICs are continuous and differentiable. Although the method developed in this paper is not restricted on how the derivatives of the SVICs are obtained, the process in which the time derivatives are computed in this work is described as follows.

To avoid having to derive higher-order time derivatives analytically, algorithmic differentiation techniques and the properties of the chain rule are exploited to automate the process. Figure 2 provides a schematic of how the higher-order time derivatives of the active SVICs are obtained. The algorithmic differentiation software *ADiGator* [43] is used to obtain the partial derivatives of the active SVIC with respect to both the state and time. To complete the full time derivative a stand-alone function, $s(\mathbf{y}(t), t)$, is written by applying the chain rule and properly substituting the dynamic equations for $d\mathbf{y}(t)/dt$. The algorithm then checks if the stand-alone function is now a function of any control variables. If the stand-alone function is a function of any of the control variables then the algorithm terminates. Otherwise, the stand-alone function re-enters the algorithm loop and is treated as if it were the original SVIC (given it is not yet a function of any of the control variables). Upon completion the derivative count q defines the order of the SVIC and is used when defining the additional necessary conditions given by Eqs. (40) and (43). Note, the process shown in Fig. 2 only needs to be executed once and can be done before solving the problem.

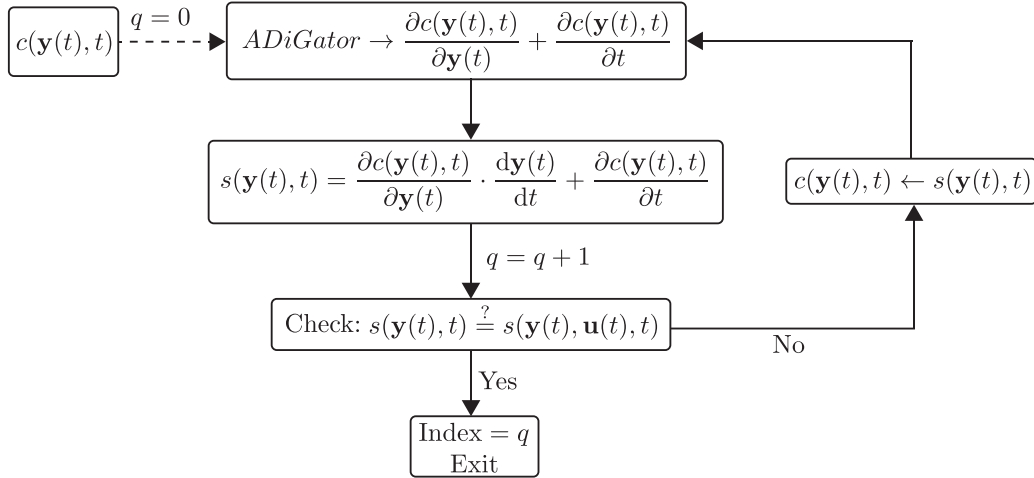


Figure 2: Schematic of algorithm for obtaining higher-order time derivatives of any active state-variable inequality path constraint.

5.2.2 Regular Domain Refinement

Suppose now that the solution $\mathbf{Y}^{(M)}$ obtained after performing structure detection and enforcing the additional necessary conditions satisfies the condition $\max\{\Delta c_i^{(M)}\} \leq \epsilon_c$, that is, the maximum constraint violations are within the user specified constraint satisfaction tolerance. Although the constraints have met the specified tolerance, certain domains may still require mesh refinement. That is, the maximum relative error $e_{\max}^{(M)}$ from the solution $\mathbf{Y}^{(M)}$ exceeds the specified mesh error tolerance. To date, several mesh refine-

ment methods have been developed [12–16]. For the problems solved in this work, the domains which did not meet the mesh tolerance were refined using the mesh refinement method developed in Ref. [16]. The process for which Ref. [16] computes the maximum relative error follows the same process as shown in Sect. 3.4. The reader is referred to Ref. [16] for a detailed explanation as to how the mesh refinement method operates.

5.2.3 Constrained Domain Refinement

Suppose $\mathbf{Y}^{(M)}$ is the solution after solving the reformulated NLP obtained from performing the structure detection and decomposition method of Sect. 5.1 and enforcing the additional necessary conditions as given in Sect. 5.2.1. Suppose now that the computed maximum constraint violation

$$\max\{\Delta c_i^{(M)}\} = \max\{c_i(\mathbf{Y}^{(M)}) - c_{i,\max}\} \quad (44)$$

on the current mesh M is larger than the user specified constraint satisfaction tolerance ϵ_c , that is, the condition $\max\{\Delta c_i^{(M)}\} \geq \epsilon_c$ holds. Suppose further that the maximum relative error on the current mesh $e_{\max}^{(M)}$ does not satisfy the specified mesh tolerance ϵ_{mesh} , that is, the condition $e_{\max}^{(M)} > \epsilon_{\text{mesh}}$ holds. It may be the case that the reformulated NLP is ill-conditioned and/or the grid on which the problem was previously solved is too coarse. Ill-conditioning can result from enforcing the additional necessary conditions at incorrect locations throughout the trajectory, which in this work would imply the detected structure could be inaccurate. To account for this possibility, the next step taken is to re-perform structure detection using the solution $\mathbf{Y}^{(M)}$. Additionally, given that the mesh error tolerance has not been met, a mesh refinement method (for example, Ref. [16]) is used to refine the previous mesh. The problem is now reformulated using the new detection results while the mesh intervals within each domain is specified in accordance with the newly refined mesh provided by the chosen mesh refinement method.

5.3 Procedure for State-Path Constrained Optimal Control Problems

An overview of the method developed in this work is provided below. The mesh iteration is denoted by M and is incremented by one upon each loop of the method. The method terminates when either a maximum prescribed mesh count, M_{\max} , is reached or when both the mesh error tolerance, ϵ_{mesh} , is satisfied on every mesh interval and the maximum constraint violation for each active SVIC, $\max\{\Delta c_i^{(M)}\}$ $i = 1, \dots, n_c$, is less than the user specified constraint satisfaction tolerance (ϵ_c).

Method for Solving State-Inequality Path Constrained Optimal Control Problems

Step 1: Set $M = 0$ and specify initial mesh. All mesh intervals form a single domain.

Step 2: Solve the NLP resulting from Radau collocation presented in Section 3.3 on mesh $M = 0$.

Step 3: Employ the structure detection and decomposition method of Section 5.1.

- (a): Determine activation/deactivation times in each constraint using the method of Section 5.1.1.
- (b): Partition the time horizon into domains using the method of Section 5.1.2.
- (c): Classify the domains as either constrained or unconstrained.
- (d): Enforce additional necessary conditions as described in Section 5.2.1.

Step 4: Increment $M \rightarrow M + 1$ and solve the reformulated NLP.

Step 5: Compute maximum constraint violations, $\max\{\Delta c_i^{(M)}\}$, and maximum relative error, $e_{\max}^{(M)}$, on current mesh.

Step 6: If $(\max\{\Delta c_i^{(M)}\} \leq \epsilon_c \text{ and } e_{\max}^{(M)} \leq \epsilon_{mesh})$ or $(M > M_{\max})$, then quit. Otherwise,

- (a): If $\max\{\Delta c_i^{(M)}\} \geq \epsilon_c$,
 - (i): Perform domain refinement according to Section 5.2.3
 - (ii): Return to **Step 3**.
- (b): If $\max\{\Delta c_i^{(M)}\} \leq \epsilon_c$, and $e_{\max}^{(M)} > \epsilon_{mesh}$,
 - (i): Perform domain refinement according to Section 5.2.2
 - (ii): Return to **Step 4**.

6 Examples

In this section, the aforementioned SPOC method described in Sect. 5 is demonstrated on two examples. The first example consists of a second-order state-variable inequality path constraint with a known analytical solution which provides a baseline to analyze the accuracy of the proposed method. The second example involves a problem with a first-order state-variable inequality path constraint where an analytical solution is not available and demonstrates the methods ability to improve constraint satisfaction when analyzing a nonlinear dynamical system. In both examples, the performance of the SPOC method developed in this paper will be compared against the mesh refinement method developed in Ref. [16]. Given the mesh refinement method developed in Ref. [16] is implemented within the SPOC method, comparisons were chosen to be made against this specific mesh refinement method. Additionally, while comparisons could be made against other mesh refinement methods (such as the methods of Ref. [12] and Ref. [14]), it has been found that the method of Ref. [16] typically outperforms these previously developed mesh refinement methods. Lastly, it is

noted that the method of Ref. [16] will be referred to from this point forth as the *hp*-LGR method.

All results to follow will include three solutions. The first being the solution obtained in the first two steps of the procedure given in Sect. 5.3, referred to from this point forth as the initial mesh. The second solution shown is obtained using the *hp*-LGR method and the third solution is obtained using the SPOC method. The design parameters used within the method were chosen for both examples to be: $\epsilon = 1 \times 10^{-4}$ and $\nu = 5$. The specified mesh tolerance for both the *hp*-LGR method and the SPOC method was chosen to be $\epsilon_{mesh} = 1 \times 10^{-6}$ for the first example, while for the second example, the mesh tolerance for both methods was set to $\epsilon_{mesh} = 1 \times 10^{-8}$. All results are obtained, other than those obtained using the SPOC method, using the MATLAB[®] optimal control software GPOPS – III as described in Ref. [44]. The resulting nonlinear programming problem (NLP) was solved using the NLP solver IPOPT [2] in full-Newton mode with an NLP tolerance and NLP constraint violation tolerance both set to $\epsilon_{NLP} = 1 \times 10^{-8}$. The user specified constraint satisfaction tolerance for both examples was set equal to the NLP tolerance, $\epsilon_c = \epsilon_{NLP}$. The first and second derivatives supplied to IPOPT were obtained using the built-in sparse finite central difference method in GPOPS – III which uses the method of Ref. [45]. In both examples, the initial mesh consisted of ten mesh intervals with four collocation points within each mesh interval. The initial guess for the initial mesh consisted of a straight line connecting boundary conditions for states with boundary conditions specified at both the beginning and end of the original time domain, while a constant value was supplied for states with a single boundary condition specified at either the end of the original time domain. Finally, all numerical computations were performed on a 2.9 GHz Intel Core i9 MacBook Pro running macOS Monterey Version 12.5 with 32 GB 2400 MHz DDR4 of RAM, using MATLAB[®] version R2021a (build 9.10.0.1669831).

6.1 Example 1: Bryson-Denham Problem

Consider the following optimal control problem obtained from Ref. [21], often referred to as the Bryson-Denham problem, which includes a second-order state-variable inequality path constraint. Minimize the objective functional,

$$\mathcal{J} = \frac{1}{2} \int_0^1 u^2(t) dt, \quad (45)$$

subject to the dynamic constraints,

$$\begin{aligned} \dot{x}(t) &= v(t), \\ \dot{v}(t) &= u(t), \end{aligned} \quad (46)$$

the boundary conditions,

$$\begin{aligned} x(0) &= x(1) &= 0, \\ v(0) &= -v(1) &= 1, \end{aligned} \quad (47)$$

and the state-variable inequality path constraint (SVIC),

$$x(t) \leq L, \quad (48)$$

where L is the upper limit on the SVIC. Though the problem formulation may appear rudimentary, it is challenging to obtain an accurate numerical approximation of the solution. For completeness, the analytical

solution for the control, position, and objective value to the above optimal control for $0 \leq L \leq 1/6$ is given by

$$u^*(t) = \begin{cases} -\frac{2}{3L} \left(1 - \frac{t}{3L}\right), & 0 \leq t \leq 3L \\ 0, & 3L \leq t \leq 1 - 3L \\ -\frac{2}{3L} \left(1 - \frac{1-t}{3L}\right), & 1 - 3L \leq t \leq 1 \end{cases} \quad (49)$$

$$x^*(t) = \begin{cases} L \left[1 - \left(1 - \frac{t}{3L}\right)^3\right], & 0 \leq t \leq 3L \\ L, & 3L \leq t \leq 1 - 3L \\ L \left[1 - \left(1 - \frac{1-t}{3L}\right)^3\right], & 1 - 3L \leq t \leq 1 \end{cases} \quad (50)$$

$$\mathcal{J}^* = \frac{4}{9L}. \quad (51)$$

For the results to follow, the upper limit on the SVIC was chosen to be $L = 1/8 \Rightarrow \mathcal{J}^* = 32/9$. Figure 3 shows the computed position, $x(t)$, for three solutions: the first obtained on the initial mesh, the second computed using the hp -LGR method, and the third solution computed using the SPOC method. Upon zooming in on the portion of the trajectory where the SVIC is known to be active, it is observed that both solutions obtained on the initial mesh and using the hp -LGR method do not remain on the constrained arc. Conversely, the position computed using the SPOC method is able to accurately approximate the active constraint arc. A similar comparison is given in Fig. 4 for the time-history of the computed optimal control,

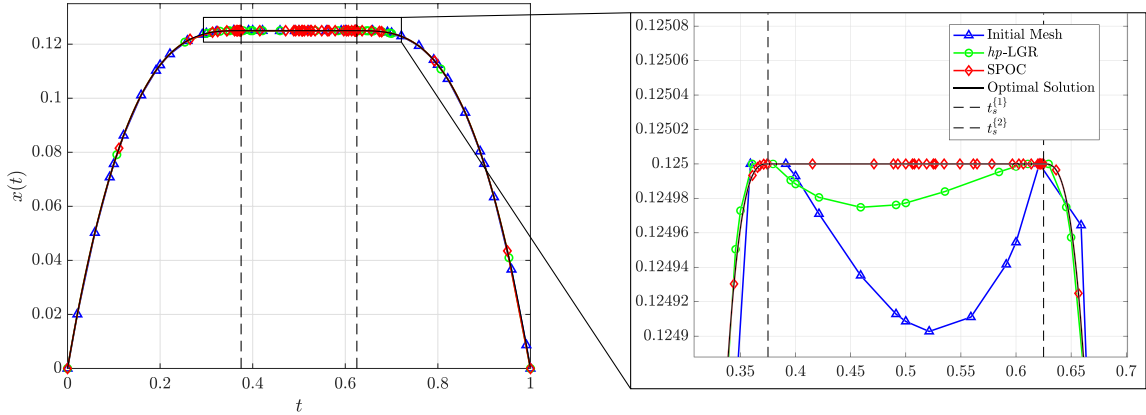


Figure 3: Position, $x(t)$, comparison between the solution obtained on the initial mesh, with the hp -LGR method of Ref. [16], and the SPOC method for the Bryson-Denham problem.

$u(t)$. Specifically, upon zooming in on the region where the constraint is known to be active, Fig. 4 shows that the optimal control computed on the initial mesh and with the hp -LGR method exhibits "chattering" behavior near the activation and deactivation of the SVIC. Conversely, the optimal control computed by the

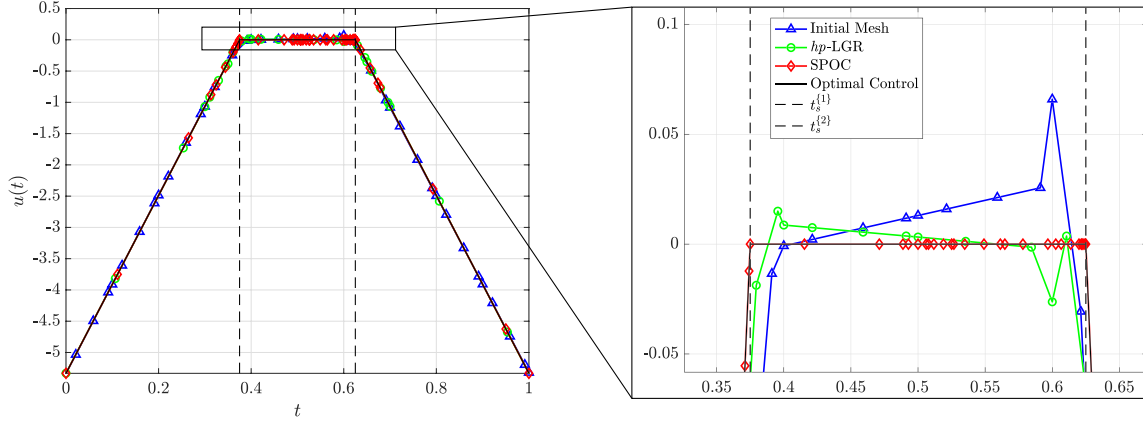


Figure 4: Control, $u(t)$, comparison between the solution obtained on the initial mesh, with the hp -LGR method of Ref. [16], and the SPOC method for the Bryson-Denham problem.

SPOC method removes any chattering behavior and closely approximates the known optimal control along the active constraint arc.

Using a detection tolerance of $\epsilon = 1 \times 10^{-4}$, and a value of $\nu = 5$, Table 1 provides the estimated activation and deactivation (A/D) times (\hat{t}_s) using the detection algorithm presented in Sect. 5.1.1. Also provided in Table 1 are the lower and upper bounds (\hat{t}_s^-, \hat{t}_s^+) on the detected times, respectively, the optimized A/D times (t_s^*) returned by the NLP solver, and the known analytical switch times. By comparing the estimated and optimized switch times with the analytical switch times, it is observed that the NLP solver is capable of selecting switch times which are in close agreement to the known analytical A/D times. This result shows that given the provided search space for the A/D times contains the optimal switch time, the NLP solver is capable of optimizing the location of the A/D times such that it is in close agreement with the analytical solution. Note, the dashed lines shown in both Figs. 3 and 4 represent the activation and deactivation times returned by the NLP. Table 2 provides the CPU times and computed relative errors with respect to

Switch time	\hat{t}_s	\hat{t}_s^-	\hat{t}_s^+	t_s^*	Analytic
Activation	0.359053	0.169957	0.519493	0.374977	0.3750
Deactivation	0.659053	0.469957	0.819493	0.625011	0.6250

Table 1: Detection performance: Estimated A/D times, \hat{t}_s , lower and upper bounds \hat{t}_s^-, \hat{t}_s^+ , optimized A/D times t_s^* , and the analytical A/D times.

the known analytical solution in the objective value, $\delta\mathcal{J}$, the activation time, $\delta t_s^{[1]}$, and the deactivation time, $\delta t_s^{[2]}$, for each solution shown in Figs. 3 and 4. Additionally, Table 2 includes a comparison against the Uniform Trigonometrization Method (UTM) (developed in Ref. [35]) which follows the procedures of a penalty method within an indirect formalism. It is found that the SPOC method computes an objective that

is eight orders of magnitude more accurate than the objective obtained using the *hp*-LGR method, and up to nine orders of magnitude of improvement with respect to the objective given in Ref. [35]. Additionally, an improvement of up to four orders of magnitude in the computed A/D times is obtained when using the SPOC method.

	$\delta\mathcal{J}$	$\delta t_s^{\{1\}}$ [s]	$\delta t_s^{\{2\}}$ [s]	CPU Time [s]
Initial Mesh	4.194475×10^{-5}	3.629784×10^{-2}	1.142003×10^{-2}	0.32
<i>hp</i> -LGR	9.716961×10^{-6}	2.884690×10^{-2}	1.785373×10^{-2}	0.40
UTM [35]	3.781250×10^{-4}	–	–	–
SPOC	3.023831×10^{-13}	2.033561×10^{-5}	8.851712×10^{-6}	2.36

Table 2: Comparison of the CPU time and the relative error of the computed cost, $\delta\mathcal{J}$, the activation time, $\delta t_s^{\{1\}}$, and the deactivation time, $\delta t_s^{\{2\}}$, with respect to the analytical solution for the initial mesh, *hp*-LGR method, the UTM method [35], and the SPOC method.

6.2 Example 2: Constrained Entry Vehicle Crossrange Maximization

Consider the following constrained reusable launch vehicle reentry optimal control problem, obtained from Ref. [46], of maximizing the achieved crossrange during atmospheric entry while being subject to a stagnation point heating rate constraint. Minimize the cost functional,

$$\mathcal{J} = -\phi(t_f) \quad (52)$$

subject to the dynamic constraints,

$$\dot{r} = v \sin \gamma, \quad (53a)$$

$$\dot{\theta} = \frac{v \cos \gamma \sin \psi}{r \cos \phi}, \quad (53b)$$

$$\dot{\phi} = \frac{v \cos \gamma \cos \psi}{r}, \quad (53c)$$

$$\dot{v} = -D - g \sin \gamma, \quad (53d)$$

$$\dot{\gamma} = \frac{L \cos \sigma}{v} + \cos \gamma \left(\frac{v}{r} - \frac{g}{v} \right), \quad (53e)$$

$$\dot{\psi} = \frac{L \sin \sigma}{v \cos \gamma} + \frac{v}{r} \cos \gamma \sin \psi \tan \phi, \quad (53f)$$

the stagnation point heating rate constraint [47],

$$\dot{Q} = \hat{Q} (\rho/\rho_0)^{1/2} (v/v_c)^{3.15} \leq \dot{Q}_{\max}, \quad (54)$$

and the boundary conditions,

$$\begin{aligned}
h(t_0) &= 79.248 \text{ km} & , & \quad h(t_f) = 24.384 \text{ km}, \\
\theta(t_0) &= 0 \text{ deg} & , & \quad \theta(t_f) = \text{Free}, \\
\phi(t_0) &= 0 \text{ deg} & , & \quad \phi(t_f) = \text{Free}, \\
v(t_0) &= 7802.88 \text{ m/s} & , & \quad v(t_f) = 762.0 \text{ m/s}, \\
\gamma(t_0) &= -1 \text{ deg} & , & \quad \gamma(t_f) = -5 \text{ deg}, \\
\psi(t_0) &= 90 \text{ deg} & , & \quad \psi(t_f) = -90 \text{ deg},
\end{aligned} \tag{55}$$

where the terminal time, t_f , is free. The lift and drag specific forces are modeled, respectively, as

$$\begin{aligned}
D &= qSC_D/m, \\
L &= qSC_L/m,
\end{aligned} \tag{56}$$

with $q = \rho v^2/2$ being the dynamic pressure,

$$\rho = \rho_0 \exp(-\beta h) \tag{57}$$

is the atmospheric density, and $h = r - R_e$ is the altitude of the vehicle. The lift and drag aerodynamic coefficients are modeled, respectively, as [46],

$$\begin{aligned}
C_L &= C_{L0} + C_{L1}\alpha, \\
C_D &= C_{D0} + C_{D1}\alpha + C_{D2}\alpha^2,
\end{aligned} \tag{58}$$

The aerodynamic coefficients and physical constants used in this model are provided in Table 3, noting that Ref. [46] uses English units whereas SI units were used to obtain the results presented in this work. Given that \dot{Q} defined in Eq. (54) is always positive during atmospheric flight, it can be transformed monotonically via the natural logarithm as

$$\ell_{\dot{Q}} \equiv \log \dot{Q} = \log \hat{\dot{Q}} - \beta h/2 + 3.15 \log(v/v_c), \tag{59}$$

where the expression for the atmospheric density given in Eq. (57) was used to obtain the result shown above in Eq. (59). Then the stagnation point heating rate constraint can be rewritten as

$$\ell_{\dot{Q}} \leq \log \dot{Q}_{\max}. \tag{60}$$

It is noted in Ref. [48] that Eq. (60) behaves better computationally than the form given in Eq. (54).

For the results presented in this work, the maximum allowable stagnation point heating rate constraint was chosen to be $\dot{Q}_{\max} = 1.5 \text{ MW/m}^2$. Figure 5 provides a comparison of the computed stagnation point heating rate profiles obtained (i) on the initial mesh (ii) using the *hp*-LGR method and (iii) using the SPOC method. Zooming in on the portion where the heating rate constraint looks to be active, a couple observations can be made. The first observation is that both heating rate profiles computed using the *hp*-LGR method and on the initial mesh retain a value that remains above the prescribe constraint limit. Furthermore,

Quantity	Value	Unit
R_e	6371.20392	km
$1/\beta$	7254.24	m
ρ_0	1.22557083	kg/m ³
μ	3.98603195	m ³ /s ²
m	92079.2526	kg
S	249.909178	m ²
\hat{Q}	199.87×10^6	W/m ²
C_{L0}	-0.2070	-
C_{L1}	1.6756	1/rad
C_{D0}	0.0785	-
C_{D1}	-0.3529	1/rad
C_{D2}	2.0400	1/rad ²

Table 3: Physical and aerodynamic constants

both heating rate profiles contain some level of chattering around the activation and deactivation points. Conversely, the stagnation point heating rate profile obtained using the SPOC method is able to remove any chattering behavior around the A/D times and retains a value that remains in close agreement with the specified upper limit of $\dot{Q}_{\max} = 1.5 \text{ MW/m}^2$.

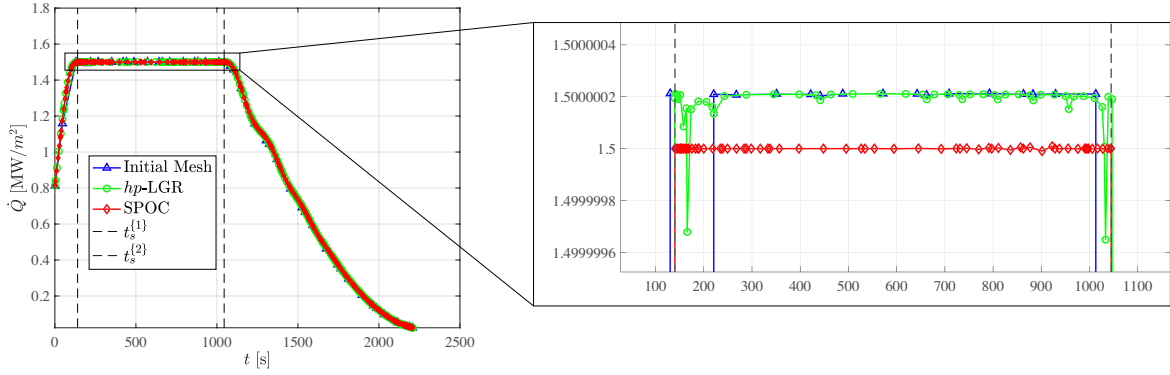


Figure 5: Heating rate profile, $\dot{Q}(t)$, comparison between the solution obtained on the initial mesh, with the hp -LGR method of Ref. [16], and the SPOC method for the constrained RLVE problem.

Figure 6a compares the computed maximum relative error (given by Eq. (33)) at each mesh iteration for both the hp -LGR method and the SPOC method. It is found that the SPOC method takes fewer mesh iterations to meet the specified error tolerance and on average obtains a maximum relative error that

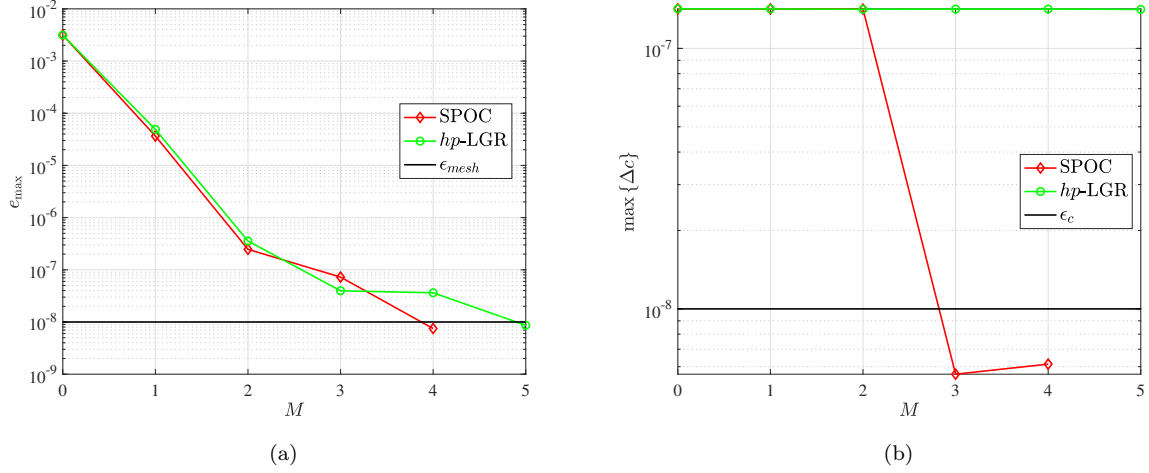


Figure 6: Maximum relative error, e_{\max} and maximum constraint violations, $\max\{\Delta c\}$ versus mesh iteration, M for the hp -LGR and SPOC methods, respectively.

is smaller than the maximum relative error obtained with the hp -LGR method at each mesh iteration. Likewise, Fig. 6b compares the computed maximum constraint violation during each mesh iteration for both the hp -LGR method and the SPOC method. It is observed that the SPOC method is capable of reducing the computed maximum constraint violation below the specified NLP tolerance while the maximum constraint violation computed using the the hp -LGR method remains approximately an order of magnitude above the specified NLP tolerance during each mesh iteration.

Table 4 provides a comparison of the computed objective values, maximum constraint violations, and CPU times for the solutions obtained (i) on the initial mesh (ii) using the hp -LGR method and (iii) using the SPOC method. First, it can be seen that the SPOC method computes an objective value that is marginally smaller than the objective obtained using the hp -LGR method. The decrease in cost is likely due to the fact that the solution obtained using the SPOC method satisfies the constraint to a tighter tolerance, and does not continuously violate the constraint throughout the constraint arc. As depicted in Fig. 6b, Table 4 re-emphasizes that the SPOC method is capable of reducing the maximum constraint violation as compared with the initial mesh and the hp -LGR method. Although the SPOC method converged to a solution in one less mesh iteration than the hp -LGR method, Table 4 shows that the SPOC method took an order of magnitude longer to compute a solution. A large percentage of the CPU time was spent within the NLP solver, and may be due to the inclusion of additional necessary conditions and decision variables.

It is emphasized that the structure detection algorithm of Sect. 5.1.1 is only performed when the maximum constraint violation is above the specified NLP tolerance, and for this example, it was executed three times before the maximum constraint violation was reduced below the specified NLP tolerance. A list of estimated and optimized A/D times obtained during the SPOC mesh iterations for which structure detection was employed is provided in Table 5. Note, estimated A/D times serve as initial guesses for the optimized

	$\mathcal{J} = \phi(t_f)[\text{deg}]$	$\max\{\dot{Q}(t) - \dot{Q}_{\max}\} [\text{MW}/\text{m}^2]$	CPU Time [s]
Initial Mesh	33.4325507	1.418×10^{-7}	1.34
<i>hp</i> -LGR	33.4465173	1.411×10^{-7}	6.40
SPOC	33.4465161	6.137×10^{-9}	50.9

Table 4: Compute cost, maximum constraint violations, and CPU time for the initial mesh, *hp*-LGR method, and the SPOC method.

A/D times which are returned by the NLP solver after solving the partitioned problem. Observing the trend in the relative error, it can be seen that the estimated and optimized A/D times converge on one another as the mesh iteration increases. This is likely due to the fact that the solution for which detection is being performed on is (i) on a denser mesh and (ii) approaching the correct constraint structure. The ability to recursively perform structure detection allows for an additional level of robustness to account for any poor/false structure detection during the initial mesh iterations where the mesh may be coarse or the approximated solution may inaccurately represent the active constraint arc.

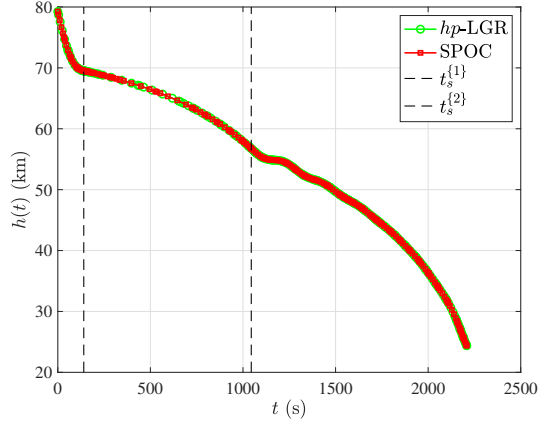
	Activation time [s]			Deactivation time [s]			Stopping criteria
Mesh Iteration	\hat{t}_1	t_1^*	$\frac{ \hat{t}_1 - t_1^* }{t_1^*} \times 100$	\hat{t}_2	t_2^*	$\frac{ \hat{t}_2 - t_2^* }{t_2^*} \times 100$	$\max\{\Delta c\} \leq \epsilon_c$
1	220.66	150.68	46.44 %	1012.96	1048.18	3.360 %	No
2	135.26	144.18	6.19 %	1062.67	1046.80	1.516 %	No
3	138.85	139.52	0.48 %	1046.80	1046.75	0.005 %	Yes

Table 5: Estimated, \hat{t}_s , optimized, t_s^* , and relative difference of the activation and deactivation times obtained throughout mesh iterations involving structure detection for Example 2.

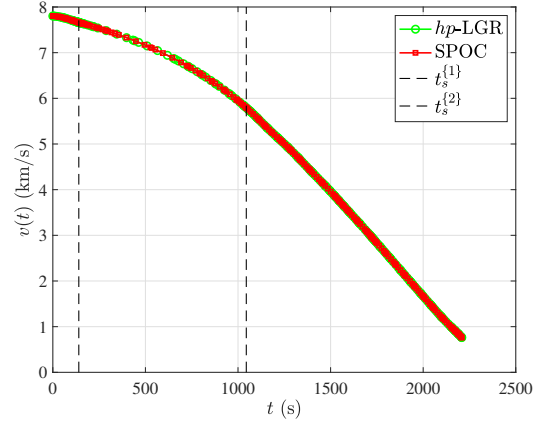
For completeness, the state and control solutions obtained with both the *hp*-LGR and SPOC methods is provided in Figure 7. The solutions are found to be in close agreement with one another, which is expected given the computed objective values are in close agreement with one another.

7 Limitations of the SPOC Method

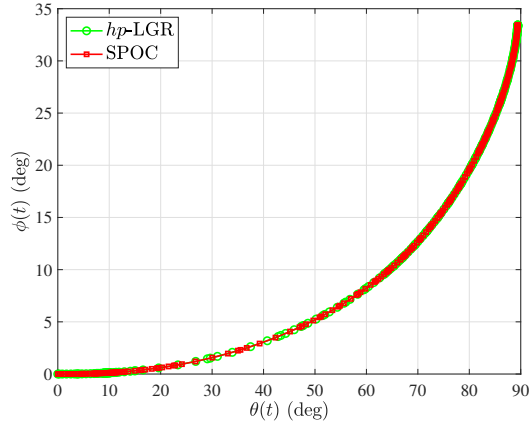
Like most computational methods, some trade-offs and limitations exist with the SPOC method. The most apparent trade-off that exists within the SPOC method is between accuracy and computational time. Both examples presented in this work display an improvement in accuracy as compared against the *hp*-LGR method, but at the expense of a larger computation time. One such limitation to the SPOC method is the assumption that the functions defining the state-variable inequality path constraints are continuous and



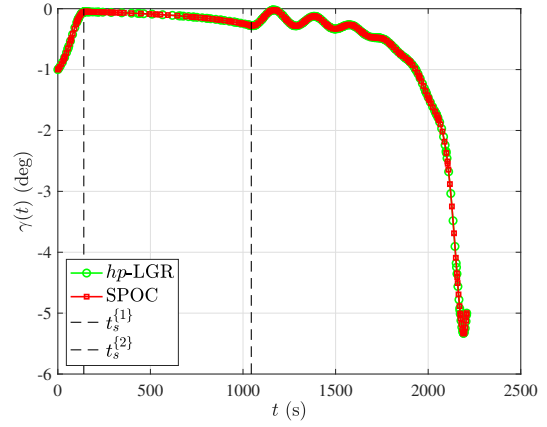
(a) Altitude, $h(t)$ vs. time, t



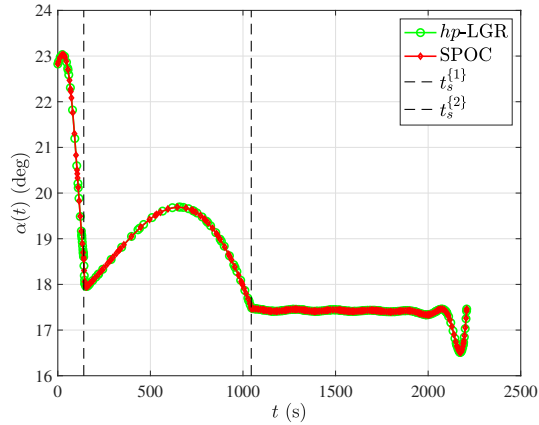
(b) Speed, $v(t)$ vs. time, t



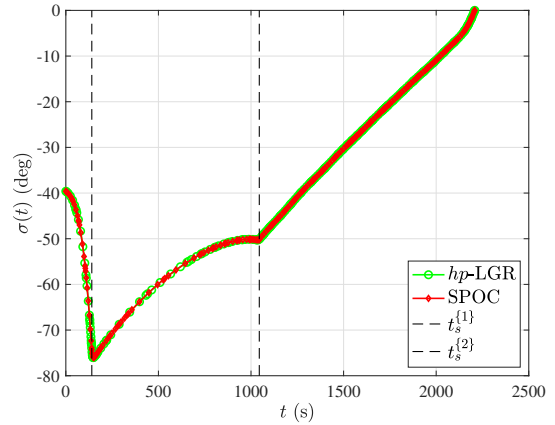
(c) Latitude, $\phi(t)$ vs. longitude, $\theta(t)$



(d) Flight path angle, $\gamma(t)$, vs. time, t



(e) Angle of attack, $\alpha(t)$ vs. time, t



(f) Bank angle, $\sigma(t)$ vs. time, t

Figure 7: Solution to Example 2 for both the hp -LGR and SPOC methods, respectively.

differentiable, which may not always be the case. For example, one might have a set of tabular data that defines the upper limits on a particular state-variable inequality path constraint. If this is the case, the SPOC method would require the user to perform some form of a function approximation leading to a potential loss of accuracy in capturing the true behavior of the constraint.

Another limitation to the SPOC method is the dependence on an initial solution and a user specified detection tolerance ϵ . Specifically, the first pass through the structure detection algorithm is dependent on the quality of the initial solution and how "tight" the detection tolerance is set. For example, decreasing the detection tolerance tightens the window for what defines an activation and deactivation point, but may lead to false activation and deactivation points if the quality of the provided solution on the previous mesh contains chattering. Conversely, increasing the detection tolerance would loosen the window for what defines an activation and deactivation point, but may lead to detecting activation and deactivation points far from the true points. While the default detection tolerance given in Sect. 5.1.1 worked for the two examples shown, it may require tuning from a user for more complex problems.

Finally, while results obtained with the SPOC method were compared against the *hp*-LGR method, it also uses the *hp*-LGR method within to suggest new meshes when the maximum relative error hasn't been met. A potential direction for future work could investigate a refinement algorithm tailored towards problems with state-variable inequality path constraints, and even problems with other types of constraints such as control constraints and mixed-state control constraints.

8 Conclusions

A method for solving optimal control problems with state-variable inequality path constraints has been developed. The method implements a detection algorithm that returns estimated activation and deactivation times of the active state-variable inequality path constraints. Based on the detection results, the method partitions the original time domain into multiple subdomains where the constraints are deemed to either be active or inactive. Within the multiple-domain formulation, the method introduces additional decision variables in the optimization process which represent the estimated activation and deactivation times, thus allowing for the exact locations of the activation and deactivation times to be optimized. For domains categorized as constrained, the lowest time derivative of the active path constraints which are explicit in a control variable is set to zero, while the preceding derivatives and the constraints themselves are set to zero at the beginning of the constrained domain. Lastly, two stopping criteria are checked to determine if an adaptive mesh refinement procedure must be employed and/or the structure detection algorithm must be used to re-detect the active constraints. The method has been demonstrated on two examples from the existing literature and compared against the method of Ref. [16]. It is found that the method is capable of producing an accurate solution to optimal control problems involving state-variable inequality path constraints.

9 Acknowledgments

The authors gratefully acknowledge support for this research from the U.S. National Science Foundation under grant CMMI-2031213, the U.S. Office of Naval Research under grant N00014-22-1-2397, and from the U.S. Air Force Research Laboratory under contract FA8651-21-F-1041.

References

- [1] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Review*, vol. 47, pp. 99–131, January 2002.
- [2] L. T. Biegler and V. M. Zavala, “Large-Scale Nonlinear Programming Using IPOPT: An Integrating Framework for Enterprise-Wide Optimization,” *Computers and Chemical Engineering*, vol. 33, pp. 575–582, March 2008.
- [3] R. H. Byrd, J. Nocedal, and R. A. Waltz, “Knitro: An integrated package for nonlinear optimization,” *Large Scale Nonlinear Optimization*, pp. 35–39, Springer Verlag, 2006.
- [4] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, “Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method,” *Journal of Guidance, Control, and Dynamics*, vol. 29, pp. 1435–1440, November-December 2006.
- [5] A. V. Rao, D. A. Benson, C. L. Darby, C. Francolin, M. A. Patterson, I. Sanders, and G. T. Huntington, “Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method,” *ACM Transactions on Mathematical Software*, vol. 37, April–June 2010, Article 22, 39 pages.
- [6] S. Kameswaran and L. T. Biegler, “Convergence Rates for Direct Transcription of Optimal Control Problems Using Collocation at Radau Points,” *Computational Optimization and Applications*, vol. 41, no. 1, pp. 81–126, 2008.
- [7] D. Garg, M. A. Patterson, C. L. Darby, C. Francolin, G. T. Huntington, W. W. Hager, and A. V. Rao, “Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems via a Radau Pseudospectral Method,” *Computational Optimization and Applications*, vol. 49, pp. 335–358. DOI: 10.1007/s10589-00-09291-0, June 2011.
- [8] D. Garg, W. W. Hager, and A. V. Rao, “Pseudospectral Methods for Solving Infinite-Horizon Optimal Control Problems,” *Automatica*, vol. 47, pp. 829–837. DOI: 10.1016/j.automatica.2011.01.085, April 2011.
- [9] D. Garg, M. A. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, “A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods,” *Automatica*, vol. 46, pp. 1843–1851. DOI: 10.1016/j.automatica.2010.06.048, November 2010.
- [10] G. 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, pp. 1793–1796, 1995.
- [11] H. Hou, W. W. Hager, and A. V. Rao, “Convergence of a Gauss Pseudospectral Method for Optimal Control,” in *Proceedings of the 2012 AIAA Guidance, Navigation, and Control Conference*, (Minneapolis, Minnesota), AIAA Paper 2012-4452, August 13-16, 2012.
- [12] C. L. Darby, W. W. Hager, and A. V. Rao, “An *hp*-Adaptive Pseudospectral Method for Solving Optimal Control Problems,” *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 476–502, 2011.
- [13] C. L. Darby, W. W. Hager, and A. V. Rao, “Direct Trajectory Optimization Using a Variable Low-Order Adaptive Pseudospectral Method,” *Journal of Spacecraft and Rockets*, vol. 48, pp. 433–445, May–June 2011.

- [14] M. A. Patterson, W. W. Hager, and A. V. Rao, "A ph Mesh Refinement Method for Optimal Control," *Optimal Control Applications and Methods*, vol. 36, pp. 398–421, July–August 2015.
- [15] F. Liu, W. W. Hager, and A. V. Rao, "Adaptive Mesh Refinement for Optimal Control Using Non-smoothness Detection and Mesh Size Reduction," *IEEE Transactions on Control System Technology*, vol. Published Online in Early View, pp. 4081–4106, June 2017.
- [16] F. Liu, W. W. Hager, and A. V. Rao, "Adaptive Mesh Refinement Method for Optimal Control Using Decay Rates of Legendre Polynomial Coefficients," *IEEE Transactions on Control System Technology*, vol. 26, pp. 1475–1483, July 2018.
- [17] D. Jacobson, M. Lele, and J. Speyer, "New necessary conditions of optimality for control problems with state-variable inequality constraints," *Journal of Mathematical Analysis and Applications*, vol. 35, no. 2, pp. 255–284, 1971.
- [18] S. S. L. Chang, "Optimal control in bounded phase space," *Automatica I*, pp. 55–67, January 1962.
- [19] D. O. Norris, "Nonlinear programming applied to state constrained optimization problems," *Journal of Mathematics Analysis and Applications*, vol. 43, pp. 261–272, August 1973.
- [20] J. McIntyre and B. Paiewonsky, "On optimal control with bounded state variables," *Advances in Control Systems*, 1967.
- [21] A. E. Bryson, W. F. Denham, and S. E. Dreyfus, "Optimal programming problems with inequality constraints i: Necessary conditions for extremal solutions," *AIAA Journal*, vol. 1, pp. 2544–2550, November 1963.
- [22] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. New York, NY: Interscience, 1962.
- [23] S. E. Dreyfus, "Variational problems with inequality constraints," *Journal of Mathematical Analysis and Applications*, vol. 4, pp. 297–308, 1962.
- [24] G. Leitmann, "A note on state-constrained optimal control," *Journal of Optimization Theory and Applications*, vol. 7, pp. 209–214, August 1971.
- [25] E. Kreindler, "Additional necessary conditions for optimal control with state-variable inequality constraints," *Journal of Optimization Theory and Applications*, vol. 38, pp. 241–250, 1982.
- [26] I. Russak, "On problems with bounded state variables," *Journal of Optimization Theory and Applications*, vol. 5, pp. 114–157, 1970.
- [27] I. Russak, "On general problems with bounded state variables," *Journal of Optimization Theory and Applications*, vol. 6, pp. 114–157, 1970.
- [28] I. Russak, "Relations among the multipliers for problems with bounded state constraints," *SIAM Journal of Control Optimization*, vol. 14, pp. 1151–1155, 1976.
- [29] B. Gollan, "On optimal control problems with state constraints," *Journal of Optimization Theory and Applications*, vol. 32, pp. 75–80, 1980.
- [30] R. F. Hartl, S. Sethi, and R. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Review*, vol. 37, no. 2, pp. 181–218, 1995.
- [31] N. Biehn, S. L. Campbell, L. Jay, and T. Westbrook, "Some comments on dae theory for irk methods and trajectory optimization," *Journal of Computational and Applied Mathematics*, vol. 120, no. 1-2, pp. 109–131, 2000.
- [32] W. F. Feehery and P. I. Barton, "Dynamic optimization with state variable path constraints," *Computers & Chemical Engineering*, vol. 22, no. 9, pp. 1241–1256, 1998.

- [33] J. T. Betts, N. Biehn, and S. L. Campbell, “Convergence of nonconvergent irk discretizations of optimal control problems with state inequality constraints,” *SIAM Journal of Scientific Computing*, vol. 23, no. 6, pp. 1981–2007, 2002.
- [34] P. Liu, X. Li, X. Liu, and Y. Hu, “An improved smoothing technique-based control vector parameterization method for optimal control problems with inequality path constraints,” *Journal of Optimal Control Applications and Methods*, vol. 38, pp. 586–600, aug 2016.
- [35] K. Mall, M. J. Grant, and E. Taheri, “Uniform trigonometrization method for optimal control problems with control and state constraints,” *Journal of Spacecraft and Rockets*, vol. 57, pp. 995–1007, sep 2020.
- [36] C. R. Heidrich, M. J. Sparapany, and M. J. Grant, “Generalized regularization of constrained optimal control problems,” *Journal of Spacecraft and Rockets*, jan 2021.
- [37] D. Jacobson and M. Lele, “A transformation technique for optimal control problems with a state variable inequality constraint,” *IEEE Transactions on Automatic Control*, vol. 14, pp. 457–464, 1969.
- [38] K. Graichen and N. Petit, “Incorporating a class of constraints into the dynamics of optimal control problems,” *Journal of Optimal Control Applications and Methods*, vol. 30, pp. 537–561, jan 2009.
- [39] B. C. Fabien, “Indirect solution of inequality constrained and singular optimal control problems via a simple continuation method,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, p. 021003, nov 2013.
- [40] W. F. Denham and A. E. Bryson, “Optimal programming problems with inequality constraints ii: Solution by steepest-ascent,” *AIAA Journal*, vol. 2, pp. 25–34, January 1964.
- [41] S. Kameswaran and L. T. Biegler, “Advantages of nonlinear-programming-based methodologies for inequality path-constrained optimal control problems—a numerical study,” *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 957–981, 2008.
- [42] E. R. Pager and A. V. Rao, “Method for solving bang-bang and singular optimal control problems using adaptive radau collocation,” *Computational Optimization and Applications*, pp. 857–887, jan 2022.
- [43] M. J. Weinstein and A. V. Rao, “Algorithm 984: Adigator, a toolbox for the algorithmic differentiation of mathematical functions in matlab using source transformation via operator overloading,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 44, no. 2, pp. 1–25, 2017.
- [44] M. A. Patterson and A. V. Rao, “GPOPS-II,” *ACM Transactions on Mathematical Software*, vol. 41, pp. 1–37, oct 2014.
- [45] M. A. Patterson and A. V. Rao, “Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems,” *Journal of Spacecraft and Rockets*, vol. 49, no. 2, pp. 354–377, 2012.
- [46] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.
- [47] R. Detra, N. Kemp, and F. Riddell, “Addendum to Heat Transfer of Satellite Vehicles Re-Entering the Atmosphere,” *Journal of Jet Propulsion*, December 1957.
- [48] C. L. Darby and A. V. Rao, “Minimum-fuel low-earth orbit aeroassisted orbital transfer of small spacecraft,” *Journal of Spacecraft and Rockets*, vol. 48, no. 4, pp. 618–628, 2011.