Department of Aerospace Engineering

Indian Institute of Technology, Bombay

Direct Methods for Optimal Control

AE 419: Supervised Learning Project

by

Nakul Randad (180010047)

under the supervision of

Prof. Debasish Chatterjee

May 8, 2021

Certificate

I certify that this Supervised Learning Project Report titled "Direct Methods for Optimal Control" by "Nakul Randad" is approved by me for submission. To the best of my knowledge, the report represents the research and work carried out by the candidate. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Indian Institute of Technology, Bombay and that the research data are presented honestly and without prejudice

Date: May 8, 2021 Prof. Debasish Chatterjee

Table of Contents

Abstract

In this report, we study the numerical methods for optimal control. In particular, the direct methods are discussed with an objective to showcase an approach for general optimal control problems. These methods, which include single shooting, multiple shooting and collocation methods, are relatively simple to follow and effectively solve a wide variety of optimization problems. We illustrate each of the methods by working through example problems.

Keywords: Optimal Control, Trajectory Optimization, Direct Methods, Single Shooting, Multiple Shooting, Direct Collocation, Pseudospectral Method.

Acknowledgements

I would like to express my deep gratitude to Prof Debasish Chatterjee, my research supervisor, for his patient guidance, encouragement and useful critiques throughout the duration of this project. I would also like to thank Siddhartha Ganguly, Ph.D. candidate at IITB, for the assistance in gathering resources, formatting the report and the valuable discussions pertaining to the topic.

List of Figures

Chapter 1

Introduction

Optimal control theory is a branch of mathematical optimization that deals with finding a control for a dynamical system over a period of time such that an objective function is optimized. Because of the complexity of most applications, optimal control problems are most often solved numerically.

1.1 Problem Formulation

In general, an objective function can include two terms: a boundary objective $\Phi(\cdot)$ and a path integral along the entire trajectory, with the integrand $\mathscr{L}(\cdot)$. A problem with both terms is said to be in *Bolza form*. A problem with only the integral term is said to be in *Lagrange form*, and a problem with only a boundary term is said to be in *Mayer form* [5].

$$J := \Phi\big(x(t_0), t_0, x(t_f), t_f\big) \ + \ \int_{t_0}^{t_f} \mathscr{L}\big(x(t), u(t), t\big) \ dt \tag{1.1}$$

Here we determine the *state* (equivalently, the trajectory), $x(t) \in \mathbb{R}^n$, the *control* $u(t) \in \mathbb{R}^m$, the *initial time*, $t_0 \in \mathbb{R}$, and the *terminal time*, $t_f \in \mathbb{R}$ (where $t \in [t_0, t_f]$ is the independent variable) that optimizes (*generally minimize*s) the **objective function** J in (1.1). The optimization is subject to a variety of limits and constraints, detailed in (1.2) – (1.4).

The first, and perhaps most important, of these constraints is the **system dynamics**, which are typically nonlinear and describe how the system changes in time:

$$\dot{x}(t) = f(t, x(t), u(t)) \tag{1.2}$$

Where $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ is a continuously differentiable map in x, u and t which constitutes the dynamics corresponding to (1.2). Next is the **path constraint**, which enforces restrictions along the trajectory.

$$h(t, x(t), u(t)) \leq 0 \tag{1.3}$$

Another important type of constraint is a nonlinear **boundary constraint**, which puts restrictions on the initial and final states of the system.

$$g(t_0, t_f, x(t_0), x(t_f)) \leq 0 \tag{1.4}$$

Numerical methods for solving optimal control problems are divided into two major classes: indirect methods (*optimize, then discretize*) and direct methods (*discretize, then optimize*) [1].

1.2 Indirect Methods

Indirect Methods use the necessary conditions of optimality of the infinite problem to derive a *boundary value problem* (BVP) in *ordinary differential equations* (ODE). The BVP is numerically solved to determine candidate optimal trajectories called extremals. Each of the computed extremals is then examined to see if it is a local minimum, maximum, or a saddle point. Of the locally optimizing solutions, the particular extremal with the lowest cost is chosen.

Here we seek a solution to the (closed system of) conditions of optimality which can be derived using the well-known calculus of variations and the Euler-Lagrange differential equations, and *Pontryagin's maximum principle* (a necessary condition), or by solving the *Hamilton–Jacobi–Bellman equation* (a sufficient condition) [6].

1.3 Direct Methods

Direct methods to continuous optimal control finitely parameterize the infinite dimensional decision variables, notably the controls $u(t)$ and sometimes both state $x(t)$ and controls $u(t)$, such that the original problem is approximated by a finite dimensional nonlinear program (NLP). This NLP can then be addressed by structurally exploiting numerical NLP solution methods. For this reason, the approach is often characterized as “First discretize, then optimize.” We will discuss *direct single shooting*, *direct multiple shooting*, and *direct collocation* in this report.

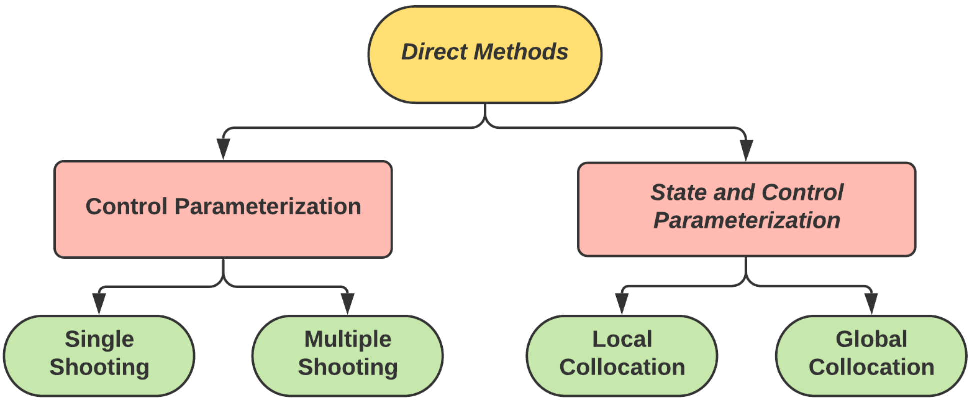


Figure 1.1: Different types of Direct Methods

One of the most important advantages of direct compared to indirect methods is that they can easily treat inequality constraints, like the inequality path constraints in the formulation above. This is because structural changes in the active constraints during the optimization procedure are treated by well developed NLP methods that can deal with inequality constraints and active set changes. All direct methods are based on a finite dimensional parameterization of the control trajectory, but differ in the way the state trajectory is handled.

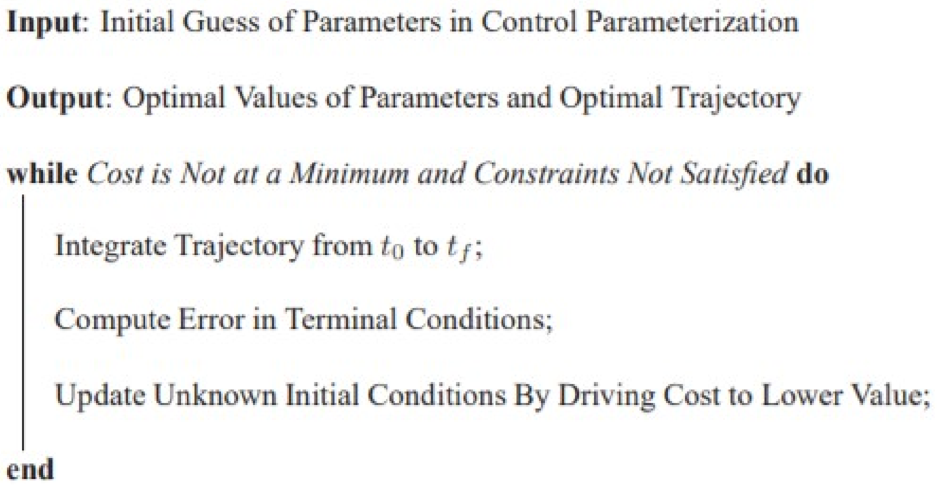
Chapter 2

Direct Shooting Methods

2.1 Overview

All shooting methods use an embedded *ordinary differential equation* (ODE) or *differential-algebraic equation* (DAE) solver in order to eliminate the continuous time dynamic system. They do so by first parameterizing the control function $u(t)$, e.g. by polynomials, by piecewise constant functions, or, more generally, by piecewise polynomials. [1]

2.2 Single Shooting



```
Input: Initial Guess of Parameters in Control Parameterization
Output: Optimal Values of Parameters and Optimal Trajectory
while Cost is Not at a Minimum and Constraints Not Satisfied do
    Integrate Trajectory from $t_0$ to $t_f$;
    Compute Error in Terminal Conditions;
    Update Unknown Initial Conditions By Driving Cost to Lower Value;
end
```

Figure 2.1: Basic algorithm for single shooting: ***ref. [1]***

We denote the finite control parameters by the vector q [7], and the resulting control function by $u(t;q)$. The most widespread parameterization are piecewise constant controls, for which we choose a fixed grid $0 = t_0 < t_1 < \ldots < t_N = t_f$, and N parameters $q_i \in \mathbb{R}^{n_u}$, $i = 0,\ldots,N-1$, and then we set

$$u(t;q) = q_i \quad if \quad t \in [t_i, t_i + 1].$$

Thus, the dimension of the vector $q = (q_0,\ldots,q_{N-1})$ is Nn_u. In single shooting, which is a sequential approach, we then regard the states $x(t)$ on $[t_0,t_f]$, keeping t_0 fixed at zero, as dependent variables that are obtained by a forward integration of the dynamic system, starting at x_0 and using the controls $u(t;q)$. This can be performed using methods like *Euler forward, Euler backward, Crank-Nicolson, Runge-Kutta, Hermite-Simpson* and many more depending upon the required accuracy. We denote the resulting trajectory as $x(t;q)$. In order to discretize inequality path constraints, we choose a grid, typically the same as for the control discretization, at which we check the inequalities. Thus, in single shooting, we transcribe the OCP (1.1) into the following NLP, that is visualized as

$$\begin{aligned} \min_{q\in\mathbb{R}^{Nn_u}} \quad & \Phi\big(x(t_0;q),t_0,x(t_f;q),t_f\big) \;+\; \int_{t_0}^{t_f} \mathscr{L}\big(x(t;q),u(t;q),t\big)\,\mathrm{d}t \\ \text{s.t.} \quad & h(t_i,x(t_i;q),u(t_i;q)) \leq 0 \qquad i = 0,\ldots,N-1 \\ & g(t_0,t_N,x(t_0;q),x(t_N;q)) \leq 0 \end{aligned} \tag{2.1}$$

2.3 Multiple Shooting

Multiple shooting works by breaking up a trajectory into some number of segments, and using single shooting to solve for each segment. As the segments get shorter, the relationship between the decision variables and the objective function and constraints becomes more linear. In multiple shooting, the end of one segment will not necessarily match up with the start of the next. This difference is known as a defect, and it is added to the constraint vector. Adding all of the segments will increase the number of decision variables (the start of each segment) and the number of constraints (defects).

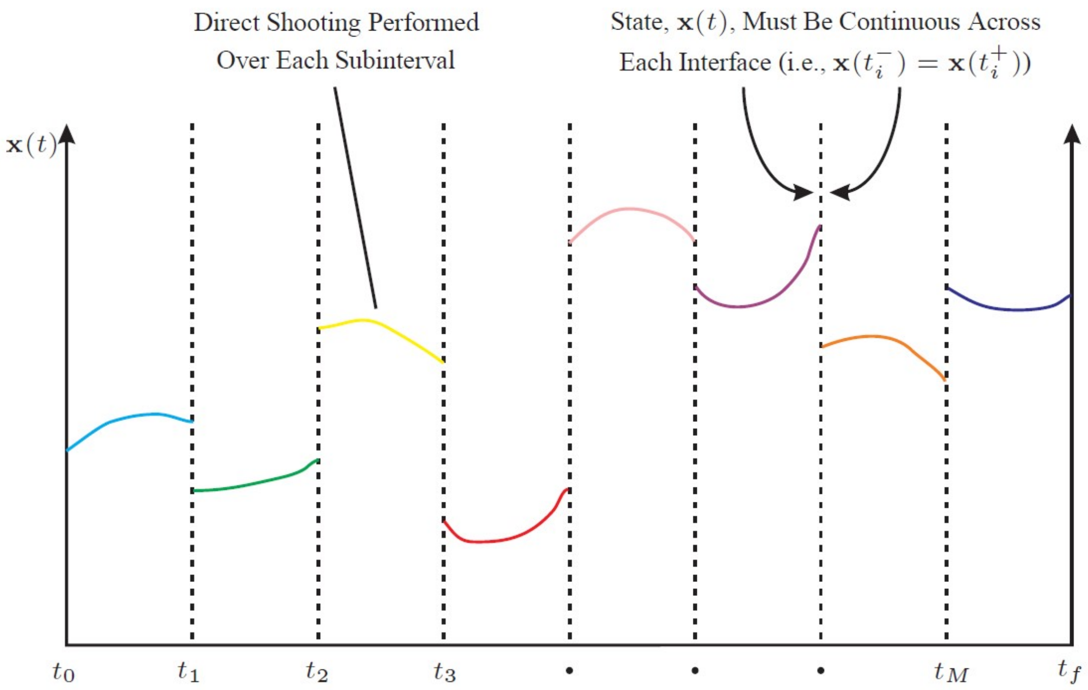

Direct Shooting Performed
Over Each Subinterval
State, $\mathbf{x}(t)$, Must Be Continuous Across
Each Interface (i.e., $\mathbf{x}(t_i^-) = \mathbf{x}(t_i^+)$)
$\mathbf{x}(t)$
t_0
t_1
t_2
t_3
•
•
•
t_M
t_f

Figure 2.2: Schematic of direct multiple shooting: ***ref. [1]***

The direct multiple shooting method performs first a piecewise control discretization on a grid, exactly as we did in single shooting, i.e. we set

$$u(t) = q_i \quad \text{for} \quad t \in [t_i, t_{i+1}].$$

But then, it solves the ODE separately on each interval $[t_i, t_{i+1}]$, starting with artificial initial values s_i:

$$\begin{aligned} \dot{x}_i(t; s_i, q_i) &= f(x_i(t; s_i, q_i), q_i) \quad \text{for} \quad t \in [t_i, t_{i+1}] \\ x_i(t_i; s_i, q_i) &= s_i. \end{aligned}$$

Thus, we obtain trajectory pieces $x_i(t; s_i, q_i)$. Likewise, we numerically compute the integrals

$$l_i(s_i, q_i) := \int_{t_i}^{t_{i+1}} \mathscr{L}\big(x_i(t; s_i, q_i), u(t; q_i), t\big) \ dt$$

Finally, we choose a grid at which we check the inquality path constraints; here we choose the same as for the controls and states, but note that a much finer sampling would be possible as well, which, however, requires continuous output from the integrator. Thus, the NLP that is solved in multiple shooting is given by:

$$\begin{aligned} \min_{s,q} \quad & \Phi(s_0, t_0, s_N, t_f) \; + \; \Sigma_{i=0}^{N-1} l_i(s_i, q_i) \\ \text{s.t.} \quad & h(t_i, s_i, q_i) \leq 0 \qquad i = 0, \ldots, N \\ & g(t_0, t_N, s_0, s_N) \leq 0 \\ & x_i(t_{i+1}; s_i, q_i) - s_{i+1} = 0 \qquad i = 0, \ldots, N-1 \end{aligned} \tag{2.2}$$

2.4 Discussions

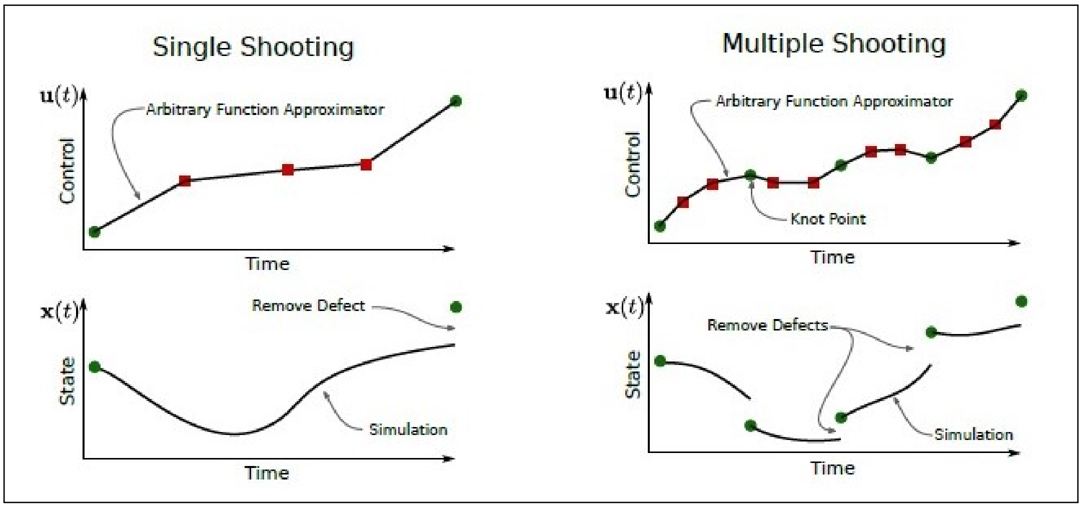


Figure 2.3: Single shooting vs multiple shooting: *ref. [2]*

In both methods the state trajectory is stored as the result of a simulation. Notice that multiple shooting is just like a series of single shooting methods, with a defect constraint added to make the trajectory continuous. Multiple shooting results in a higher dimensional non-linear program, but it is sparse and more linear than the program produced by single shooting. Therefore it is important, we can and should employ a sparsity exploiting NLP solver.

Single shooting works well enough for simple problems, but it will almost certainly fail on problems that are more complicated. This is because the relationship between the decision variables and the objective and constraint functions is not well approximated by the linear (or quadratic) model that the NLP solver uses [2]. This is not a huge problem for multiple shooting as on generating shorter segments, the relationship becomes more linear.

Chapter 3

Direct Collocation Methods

3.1 Overview

Arguably the most powerful methods for solving general optimal control problems are direct collocation methods. A direct collocation method is a state and control parameterization method where the state and control are approximated using a specified functional form. The time derivative of the state is approximated by differentiating the interpolating polynomial and constraining the derivative to be equal to the vector field at a finite set of collocation points.

The two most common forms of collocation are **local collocation** and **global collocation** [1]. The methodology classification for direct collocation is *h-methods* and *p-methods*.

3.1.1 h-methods and p-methods

Originally, direct collocation methods were developed as h-methods (for example, Euler or Runge-Kutta methods) where the time interval is divided into a mesh and the state is approximated using the same fixed-degree polynomial in each mesh interval. Convergence in an h-method is then achieved by increasing the number and placement of the mesh points. [8]

For p-methods, we divide the time horizon $[t_0, t_f]$ into (fixed) mesh intervals. The state is approximated in each interval by an N^{th} order polynomial. Convergence of the p method was then achieved by increasing the degree of the polynomial approximation. For problems whose solutions are smooth and well-behaved, a p Gaussian quadrature collocation method has a simple structure and converges at an exponential rate. The most well developed p Gaussian quadrature methods are those that employ either Legendre-Gauss (LG) points, Legendre-Gauss-Radau (LGR) points, or Legendre-Gauss-Lobatto (LGL) points. [8]

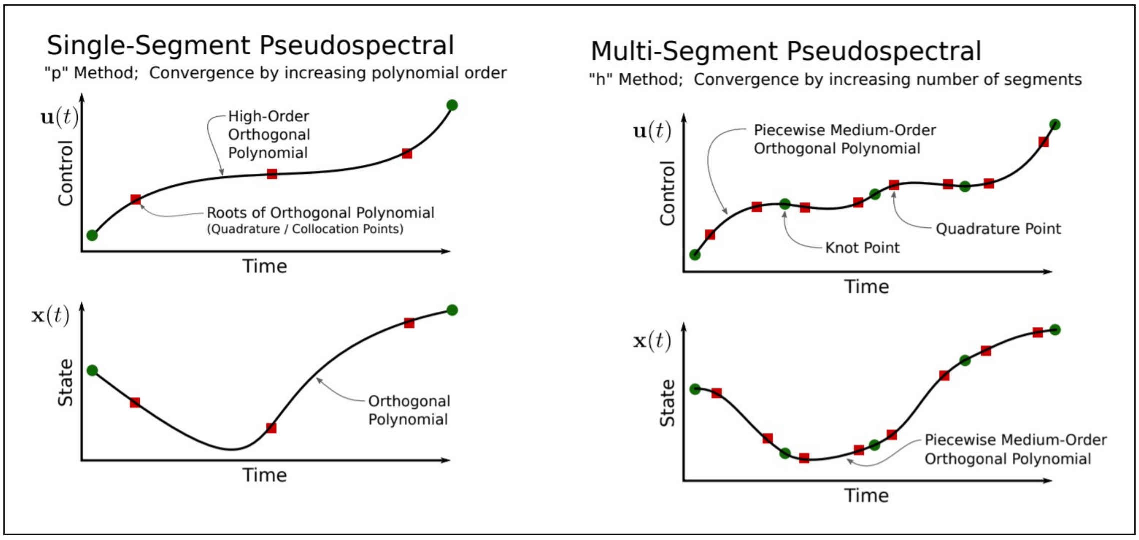

Single-Segment Pseudospectral
"p" Method; Convergence by increasing polynomial order
$\mathbf{u}(t)$
Control
High-Order Orthogonal Polynomial
Roots of Orthogonal Polynomial
(Quadrature / Collocation Points)
Time
$\mathbf{x}(t)$
State
Orthogonal Polynomial
Time
Multi-Segment Pseudospectral
"h" Method; Convergence by increasing number of segments
$\mathbf{u}(t)$
Control
Piecewise Medium-Order Orthogonal Polynomial
Knot Point
Quadrature Point
Time
$\mathbf{x}(t)$
State
Piecewise Medium-Order Orthogonal Polynomial
Time

Figure 3.1: Comparison between h-method and p-method: ref. [2]

3.2 Forms of Collocation Methods

3.2.1 Local Collocation

In a local approach, the time interval is divided into a large number of subintervals called segments or finite elements and a small number of collocation points are used within each segment. The segments are then linked via continuity conditions on the state, the independent variable, and possibly the control. The rationale for using local collocation is that a local method provides so-called local support (i.e., the discretization points are located so that they support the local behavior of the dynamics) and is both computationally simple and efficient. [4]

A local collocation method follows a procedure in which the time interval $[t_0, t_f]$ is divided into S subintervals $[t_{s-1}, t_s], \quad (s = 1, \ldots, S)$ where $t_S = t_f$. In order to ensure continuity in the state across subintervals, the following compatibility constraint is enforced at the interface of each subinterval:

$$x(t^{-i}) = x(t^{+i}), \qquad (s = 2, \ldots, S-1)$$

In the context of optimal control, local collocation has been employed using one of two categories of discretization: Runge-Kutta methods and orthogonal collocation methods. [1]

3.2.2 Global Collocation

As a global collocation method, the number of segments is set to unity (i.e., $S = 1$) and the accuracy is assessed as a function of the number of collocation points N. Thus global collocation methods need to apply p-methods using a single interval. Global polynomial methods use a high-degree global polynomial (Chebyshev or Lagrange polynomials) for parameterization, and a set of orthogonal nodes associated with the family of the polynomial for discretization.

The three most commonly used set of collocation points are Legendre-Gauss (LG), Legendre-Gauss-Radau (LGR), and Legendre-Gauss-Lobatto (LGL) points. These three sets of points are obtained from the roots of a Legendre polynomial and/or linear combinations of a Legendre polynomial and its derivatives. All three sets of points are defined on the domain $[-1, 1]$, but differ significantly in that the LG points include neither of the endpoints, the LGR points include one of the endpoints, and the LGL points include both of the endpoints. [3]

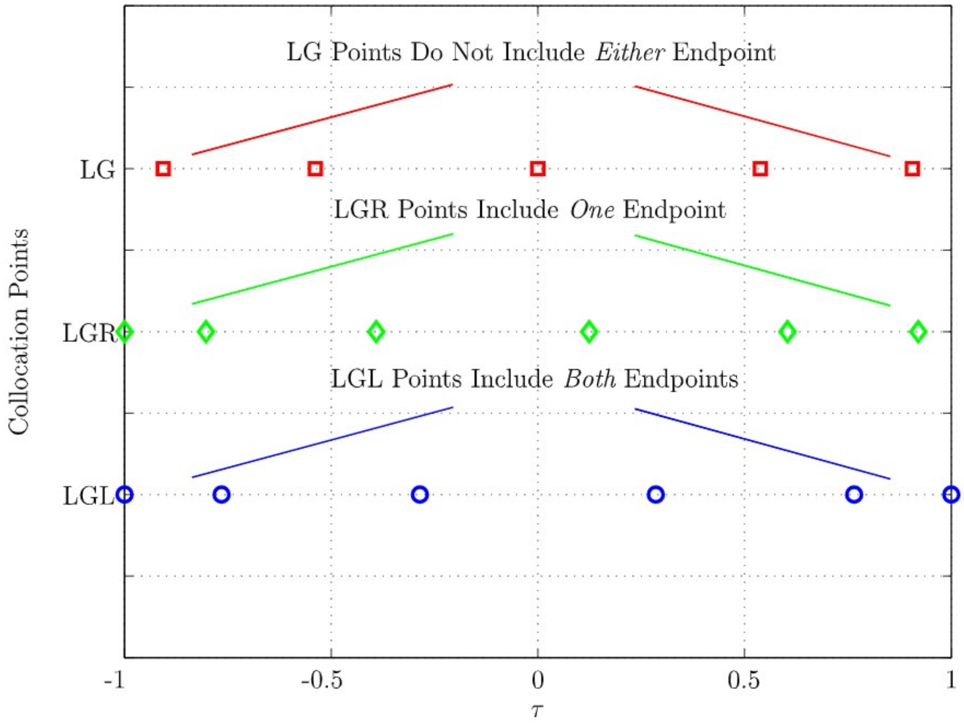


Figure 3.2: Differences between LGL, LGR, and LG collocation points: ref. [3]

3.3 Pseudospectral Method

A pseudospectral method is a form of orthogonal collocation, i.e., in a pseudospectral method the state and control are approximated using orthogonal polynomial and collocation is performed at chosen points, i.e., the collocation points are the roots of an orthogonal polynomial (or linear combinations of such polynomials and their derivatives).

Because pseudospectral methods are commonly implemented via orthogonal collocation, the terms pseudospectral and orthogonal collocation are interchangeable. One advantage to pseudospectral methods is that for smooth problems, pseudospectral methods typically have faster convergence rates than other methods, exhibiting “spectral accuracy”. [4]

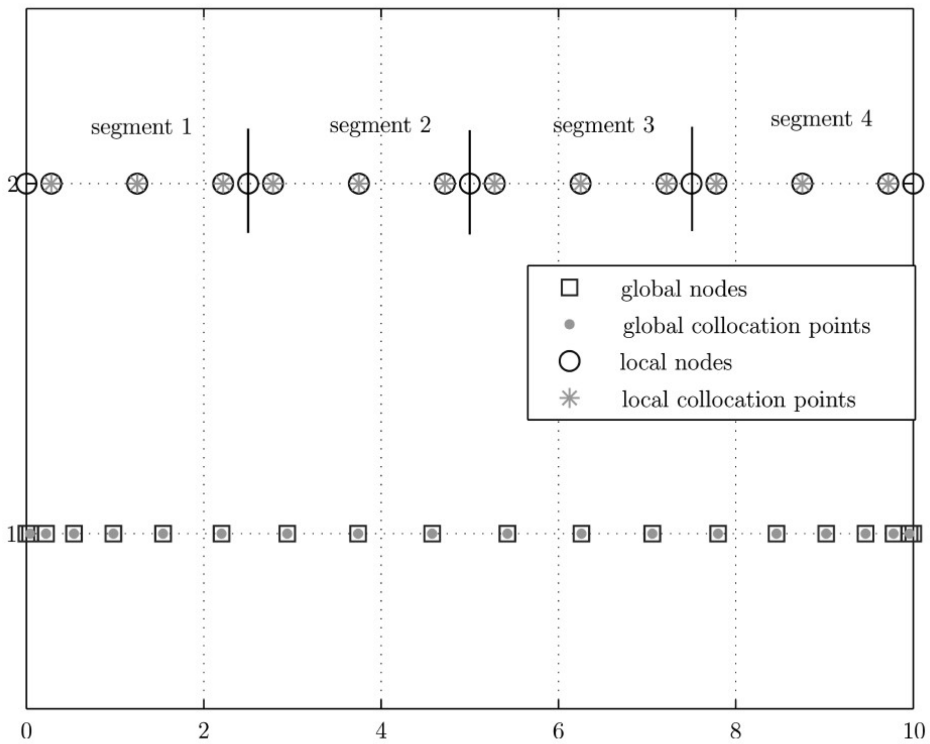


Figure 3.3: Distribution of nodes and collocation points for both the global and local approaches ($N = 20$): ref. [4]

The results obtained in [4] suggest that, except in special circumstances, global orthogonal collocation is preferable to local orthogonal collocation. [4] also suggests that, the global Gauss pseudospectral method (GPM) is much more accurate than the local GPM for a given number of total collocation points. Furthermore, for a desired accuracy, the global approach is computationally more efficient than the local approach in smooth problems. For non-smooth problems the local and global approach are quite similar in terms of accuracy.

3.4 Problem Formulation

For this illustration we employ local collocation with Legendre-Gauss (LG) collocation points. The state at a set of collocation points (in addition to the beginning of the interval) enters in the NLP as variables. An example of a choice of time points are the Legendre points of order $d = 3$:

$$\tau = [0, 0.112702, 0.500000, 0.887298]$$

Keeping the same control discretization as in single and multiple shooting:

$$u(t) = u_k, \qquad \text{for} \qquad t \in [t_k, t_{k+1}], \qquad k = 0, \ldots, N-1$$

the complete list of time points, with $h_k := t_{k+1} - t_k$, is:

$$t_{k,j} := t_k + h_k \tau_j, \qquad \text{for} \qquad k = 0, \ldots, N-1 \qquad \text{and} \qquad j = 0, \ldots, d$$

as well as the final time $t_{N,0} = t_f$. Also let $x_{k,j}$ denote the states at these time points. On each control interval, we shall define a Lagrangian polynomial basis:

$$L_j(\tau) = \prod_{r=0, r \neq j}^{d} \frac{\tau - \tau_r}{\tau_j - \tau_r}$$

Since the Lagrangian basis satisfies:

$$L_j(\tau_r) = \begin{cases} 1, & \text{if } j = r \ ; \\ 0, & \text{otherwise.} \end{cases}$$

we can approximate the state trajectory approximation as a linear combination of these basis functions:

$$\tilde{x}_k(t) = \sum_{r=0}^{d} L_r \left(\frac{t - t_k}{h_k} \right) x_{k,r}$$

In particular we get approximations of the state time derivative at each collocation point (not including τ_0):

$$\dot{\tilde{x}}_k(t_{k,j}) \quad = \quad \frac{1}{h_k} \sum_{r=0}^{d} \dot{L}_r(\tau_j)\ x_{k,r} \quad = \quad \frac{1}{h_k} \sum_{r=0}^{d} C_{r,j}\ x_{k,r}, \tag{3.1}$$

as well as an approximation of the state at the end of the control interval:

$$\tilde{x}_{k+1,0} \quad = \quad \sum_{r=0}^{d} L_r(1)\ x_{k,r} \quad = \quad \sum_{r=0}^{d} D_r\ x_{k,r}, \tag{3.2}$$

where $C_{r,j} := \dot{L}_r(\tau_j)$ and $D_r := L_r(1)$. Plugging in the approximation of the state derivative (3.1) into the ODE gives us a set of collocation equations that needs to be satisfied for every state at every collocation point:

$$h_k \ f(t_{k,j}, x_{k,j}, u_k) - \sum_{r=0}^{d} C_{r,j} \ x_{k,r} = 0, \qquad k = 0, \ldots, N-1, \qquad j = 1, \ldots, d \tag{3.3}$$

And the approximation of the end state (3.2) gives us a set of continuity equations that must be satisfied for every control interval:

$$x_{k+1,0} - \sum_{r=0}^{d} D_r \ x_{k,r} = 0, \qquad k = 0, \ldots, N-1, \tag{3.4}$$

These two sets of equations [(3.3) and (3.4)] take the place of the continuity equation (represented by the integrator call) in direct multiple shooting.

Chapter 4

Optimal Control Problems

4.1 Bryson-Denham Problem

4.1.1 Description

The Bryson-Denham optimal control problem is a benchmark test problem for optimal control algorithms. Consider the system

$$\begin{aligned} \dot{x}(t) &= v(t) \\ \dot{v}(t) &= u(t) \end{aligned} \tag{4.1}$$

The parameter $u(t) \in \mathbb{R}$ (acceleration) is adjusted over the time horizon from a starting time of zero to a final time of one. The variable $x(t) \in \mathbb{R}$ is the position and $v(t) \in \mathbb{R}$ is the velocity. Consider the optimal control problem

$$\begin{array}{ll} \text{minimize} & \displaystyle\frac{1}{2}\int_0^1 u(t)^2 \,\mathrm{d}t \\ \text{subject to} & \begin{cases} \text{underlying system dynamics (4.1)}, \\ x(0) = x(1) = 0, \\ v(0) = -v(1) = 1, \\ x(t) \leq l, \quad \text{where } l = \frac{1}{9}. \end{cases} \end{array} \tag{4.2}$$

4.1.2 Single Shooting

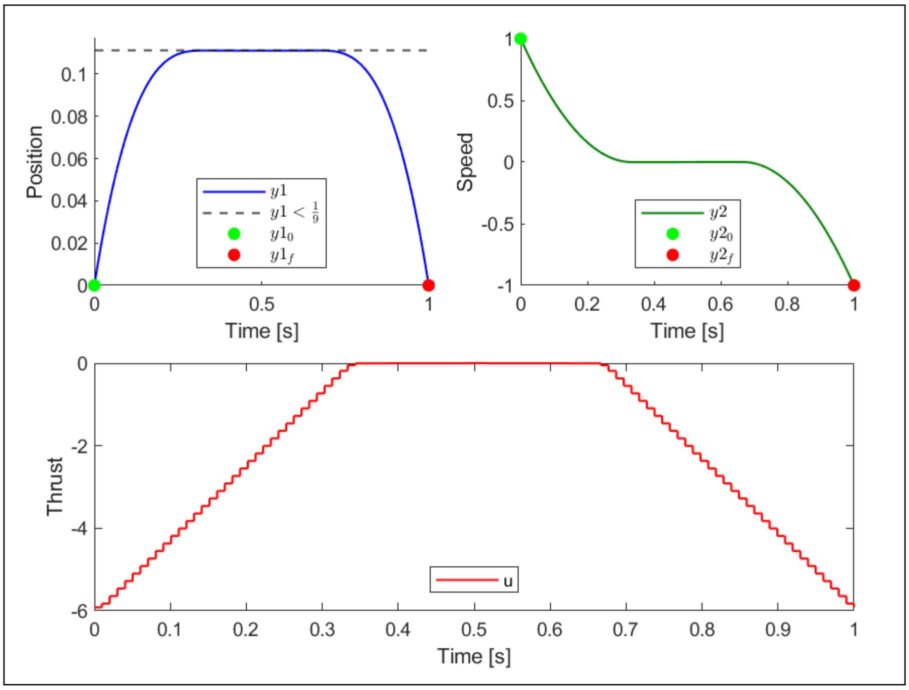


Figure 4.1: Bryson-Denham problem by single shooting

4.1.3 Multiple Shooting

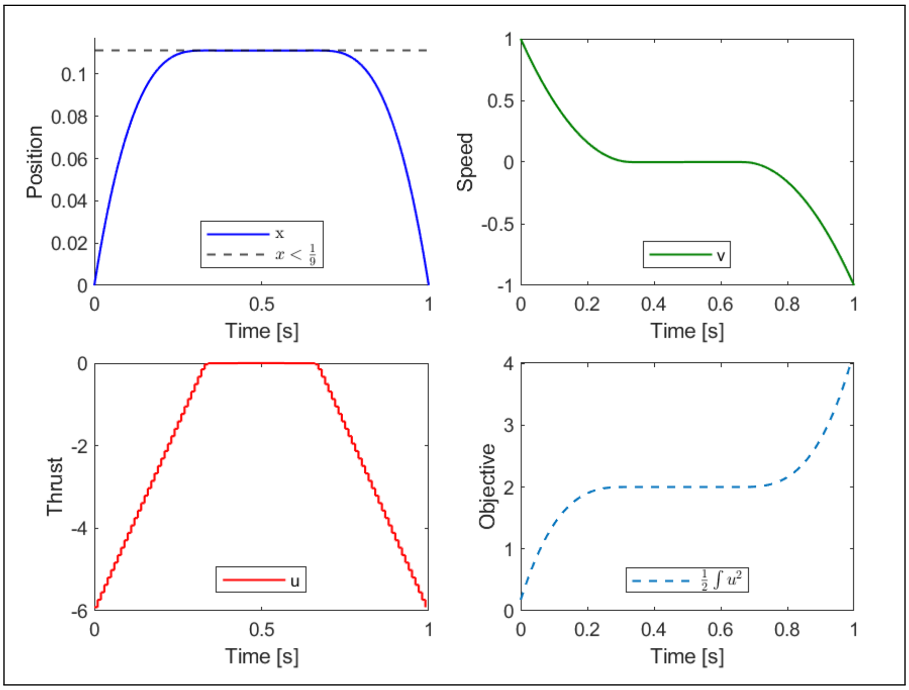


Figure 4.2: Bryson-Denham problem by multiple shooting

4.1.4 Direct Collocation

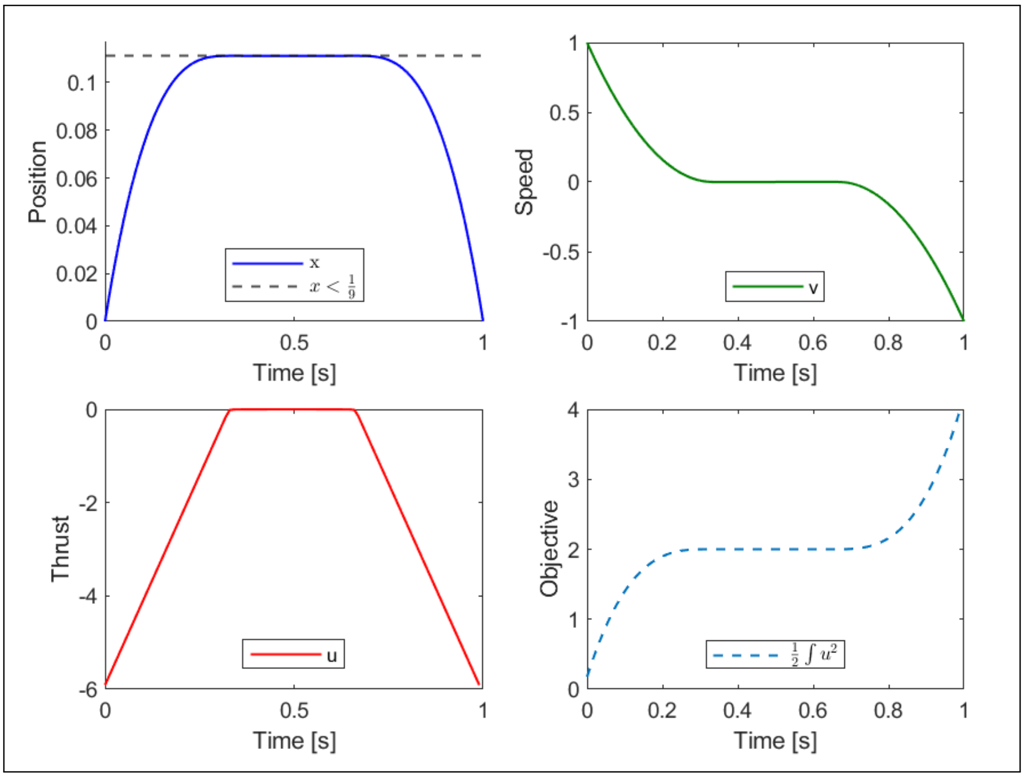


Figure 4.3: Bryson-Denham problem by direct collocation

4.2 Double Integrator Bang-Bang Problem

4.2.1 Description

Consider the system $\ddot{x} = u, \quad u \in [-1,1]$ which can represent a car with position $x \in \mathbb{R}$ and with bounded acceleration u acting as the control (negative acceleration corresponds to braking). Let us study the problem of *parking* the car at the origin, i.e., bringing it to rest at $x = 0$, in **minimal time**. It is clear that the system can indeed be brought to rest at the origin from every initial condition. However, since the control is bounded, we cannot do this arbitrarily fast (we are ignoring the trivial case when the system is initialized at the origin). Thus we expect that there exists an optimal control u^* which achieves the transfer in the smallest amount of time. [6]

We know that the dynamics of the double integrator are equivalently described by the state-space equations

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{4.3}$$

where $x_1 \in \mathbb{R}$ is the displacement of car and $x_2 \in \mathbb{R}$ is the velocity. We assume that the initial values and final values $x_1(t_0)$, $x_2(t_0)$, $x_1(t_f)$ and $x_2(t_f)$ are given (which are $(10,0,0,0)$ respectively).

4.2.2 Single Shooting

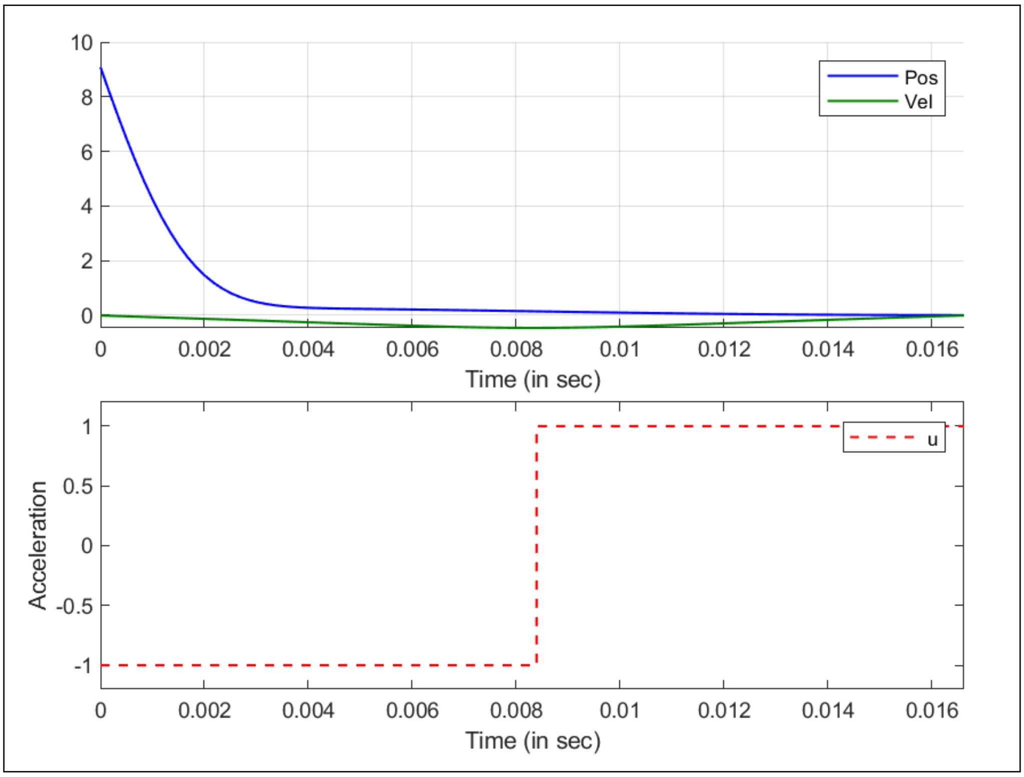


Figure 4.4: Double integrator bang-bang problem by single shooting

4.2.3 Multiple Shooting

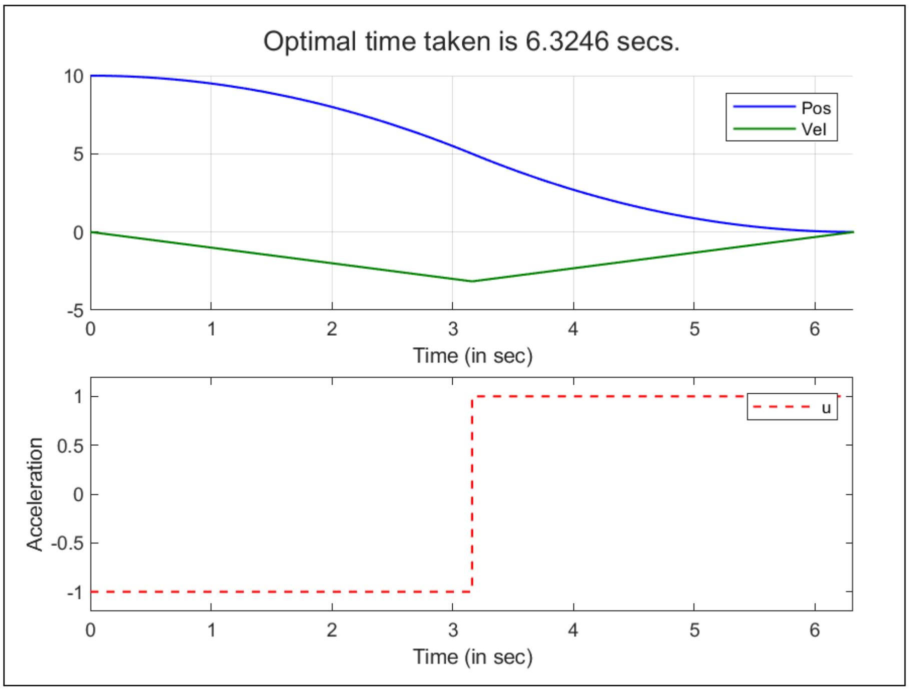


Figure 4.5: Double integrator bang-bang problem by multiple shooting

4.2.4 Direct Collocation

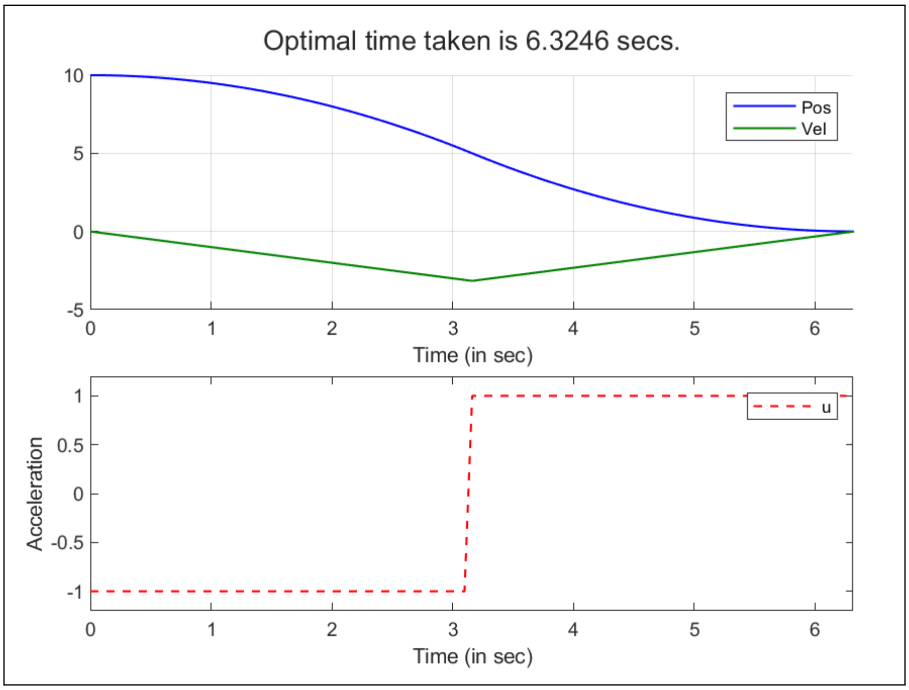


Figure 4.6: Double integrator bang-bang problem by direct collocation

4.3 Conclusions

Except for the single-shooting method used to solve bang-bang double integrator problem, all the methods gave decently correct results for each of the problems. The non-smoothness of the optimal control function in bang-bang problems might have caused single-shooting method to fail.

Looking at these direct methods for optimal control, we can summarise them in the following way: When we went from direct single shooting to direct multiple shooting we essentially traded non-linearity for problem size. The NLP in single shooting is small, but often highly nonlinear, whereas the NLP for multiple-shooting is larger, but less nonlinear and with a sparsity structure that can be exploited efficiently. Direct collocation is to take one more step in the same direction, adding even more degrees of freedom. The resulting NLP is even larger, but has even more structure that can be exploited.

Direct methods are easier to implement with effective incorporation of state and control constraints. But, this comes at the cost of accuracy and that is why it is important to choose an appropriate method based on the problem and requirements.

References

[1] Anil V. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135, 2010.

[2] Matthew Kelly. Transcription methods for trajectory optimization: a beginners tutorial. *arXiv: Optimization and Control*, 2017.

[3] Divya Garg et al. Direct trajectory optimization and costate estimation of general optimal control problems using a radau pseudospectral method. *AIAA Guidance, Navigation, and Control Conference*, 2009.

[4] Anil V. Rao Geoffrey T. Huntington. Comparison of global and local collocation methods for optimal control. *Journal of Guidance, Control, and Dynamics*, 31, 2008.

[5] Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *Society for Industrial and Applied Mathematics*, 59, 2017.

[6] Daniel Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2012.

[7] Moritz Diehl. Lecture Notes on Optimal Control and Estimation, 2014. URL: `https://syscop.de/files/2015ss/events/tempo/optimalcontrolDiehl.pdf`. Accessed: 25/04/2021.

[8] Michael. A. Patterson Anil V. Rao and William W. Hager. A ph collocation scheme for optimal control. *Optimal Control Applications and Methods*, 36(4), 2014.