Overview of Direct Methods for Dynamic Optimization—Collocation

Johan Åkesson

Dept. of Automatic Control
Lund University

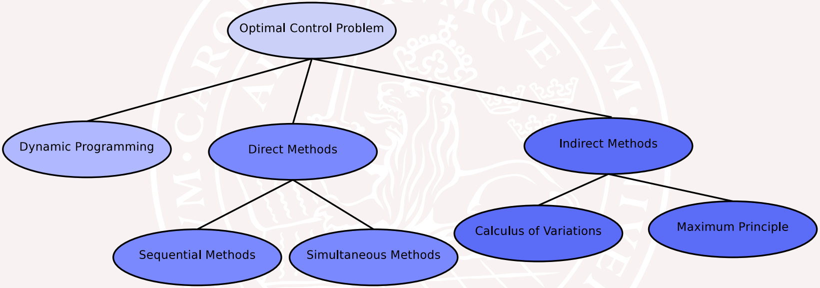
Outline

- Introduction
- Direct methods
- Simultaneous collocation methods
- Collocation based on Lagrange polynomials
- Pitfalls in dynamic optimization

(Brief) History of Optimal Control

- Calculus of Variations
  - Optimization with a differential equation as constraint
  - Minimum drag nose shape, Newton (1685)
  - The Brachistochrone problem, Bernoulli (1699)
- Goddard's rocket launch problem
  - Defined 1919
  - Solved (analytically) 1951
- The space race
  - Sputnik (1957)
  - Inter-planetary travel
  - Shuttle reentry
- Dynamic Programming, Bellman (1957)
- Maximum principle, Pontryagin (1961)
- Currently very active research area
  - Computational methods (ODEs, DAEs and PDEs)
  - On-line applications: Predictive control and estimation

Dynamic Optimization – Overview

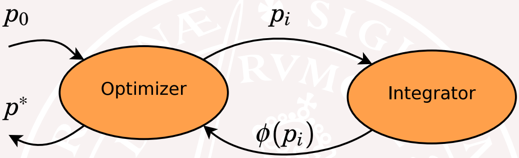


Direct Methods – Motivation

- The Maximum principle have been successfully applied in several important cases, but...
- Difficulties to derive $\frac{\partial H}{\partial x}$ and $\frac{\partial H}{\partial \lambda}$
- Path constraints difficult
  - Must know the number and order of constraint activation
- Problems with adjoint variables
  - Non-intuitive to find initial guess
  - Ill-conditioned
- Two main direct approaches
  - Simultaneous methods (Full discretization = huge NLP)
  - Sequential methods (ODE/DAE integrator + NLP solver)

Direct Shooting Methods – A Simple Approach

Connect an integrator for evaluation of $J(p)$ and an optimizer



- Very simple to implement
- Without gradients: poor convergence
- Global search methods (not gradient-based)
- Gradients—sensitivity equations

Disclaimer

Notice that this approach might be the only feasible way to go, if the system is sufficiently “difficult”, for example due to complicated hybrid behaviour.

Multiple Shooting

- Simultaneous method
- Refinement of single shooting – divide horizon into elements
  - Integrate each segment separately
- Improvements over single shooting
  - Better numerical properties due to decoupling
  - State constraints at segment junctions
- NLP larger than for single shooting but smaller than for direct collocation
- Popular in NMPC applications

Simultaneous Collocation Methods

- Motivation
  - Integration of differential equations is expensive
  - Sophisticated integrators are *accurate*, but not necessarily *consistent*
    - Very low tolerances $\Rightarrow$ long execution times
    - Noisy derivatives $\Rightarrow$ poor NLP convergence
- Basic idea of simultaneous collocation methods: *"Discretize not only the controls, but also the state variables $\Rightarrow$ problem is transcribed into discrete form in one step*
  - Not sufficient with a crude approximation: must fulfill dynamic constraint with high accuracy
  - The resulting NLP is large (but sparse)

Collocation – Introduction

Given the dynamic system

$$\dot{x} = f(x,u), \quad x(t_0) = x_0$$

a simple method for solving the differential equation is

$$\dot{x} \approx \frac{x_{k+1} - x_k}{h} \Rightarrow x_{k+1} = x_k + h f(x_k, u_k)$$

where $h = N_e/t_f$. Normally, we iterate, but what if we write all equations simultaneously?

$$\begin{pmatrix} x_1 - (x_0 + h f(x_0, u_0)) = 0 \\ x_2 - (x_1 + h f(x_1, u_1)) = 0 \\ \vdots \\ x_{N_e} - (x_{N_e-1} + h f(x_{N_e-1}, u_{N_e-1})) = 0 \end{pmatrix} \Rightarrow c(\bar{x}, \bar{u}) = 0$$

System of N_e algebraic equations to be solved for the unknowns $\bar{x} = (x_1^T, \ldots, x_{N_e}^T)^T$, if $\bar{u} = (u_1^T, \ldots, u_{N_e}^T)^T$ assumed to be known.

Collocation – Properties

- In continuous time, the differential constraint $\dot{x} = f(x, u)$ holds at *every* time instant. In the discretized formulation, the differential constraint is fulfilled only at discrete points: the *collocation points*
- Increased number of elements increases accuracy and size of NLP
- Forward Euler could be upgraded to more sophisticated one-step methods such as Runge-Kutta
- Numerical stability properties for one-step methods inherited

Collocation and Optimization

Continuous time (infinite dimensional problem)

$$\min_{u(t)} \phi(x(t_f)) \quad \text{s.t.} \quad \dot{x} = f(x,u) \quad x(0) = x_0$$

Discrete time (finite dimensional problem)

$$\min_{u_k} \phi(x_{N_e}), \quad k = 0..N_e - 1$$

subject to

$$\begin{pmatrix} x_0 + h f(x_0, u_0) - x_1 = 0 \\ \vdots \\ x_{N_e-1} + h f(x_{N_e-1}, u_{N_e-1}) - x_{N_e} = 0 \end{pmatrix} \Rightarrow c(\bar{x}, \bar{u}) = 0$$

- The *infinite* dimensional problem is transformed into a *finite* dimensional **static optimization problem**

Additional Details

- Path constraints straightforward, translates into algebraic constraint

$$c_i(x,u) \leq 0 \Rightarrow c_i(x_i,u_i) \leq 0, \quad i = 1..N_e$$
$$c_e(x,u) = 0 \Rightarrow c_e(x_i,u_i) = 0, \quad i = 1..N_e$$

- Terminal constraints are equally straightforward

$$c_t(x(t_f)) = 0 \Rightarrow c_t(x_{N_e,N_c}) = 0$$

- Minimum time problems can be formulated by optimizing also over the element lengths h_i and adding the constraint

$$\sum_{i=1}^{N_e} h_i = t_f$$

Stiff Systems and Numerical Stability

- Forward Euler is conceptually simple and easy to implement, but...
  - Numerically unstable for stiff systems
  - Requires small step length to achieve accurate solution
- One-step methods and Runge-Kutta
  - Explicit, e.g. RK45
  - Implicit RK schems has strong stability properties
- BDF-methods, strong stability properties
- Large body of results from numerical analysis applicable to collocation methods

Optimization of Differential Algebraic Systems

- Differential Algebraic Equations (DAEs)—generalized form of ODEs

$$F(p, \dot{x}, x, u, w, t) = 0, \quad t \in [t_0, t_f]$$

where $p \in R^{n_p}$ are the parameters, $\dot{x} \in R^{n_x}$ are the state derivatives, $x \in R^{n_x}$ are the states, $u \in R^{n_u}$ are the inputs and $w \in R^{n_w}$ are the algebraic variables.

- Assumptions (index-1 DAE)
  - $F \in R^{n_x+n_w}$
  - $\left|\left[\frac{\partial F}{\partial \dot{x}}, \frac{\partial F}{\partial w}\right]\right| \neq 0$
- Intuition: given x we may solve for w and $\dot{x}$ (implicit function theorem).
- Index-1 DAEs are similar to ODEs, but solution of non-linear equation systems may be needed to compute $\dot{x}$ from x

Optimization of Differential Algebraic Systems

$$\min_{p,u} J(p,q)$$

subject to

$$F(p,v) = 0,\ t \in [t_0, t_f] \qquad \text{DAE dynamics}$$

$$F_0(p,v) = 0,\ t = t_0 \qquad \text{Initial conditions}$$

$$C_{eq}(p,v,q) = 0,\ C_{ineq}(p,v,q) \leq 0,\ t \in [t_0, t_f] \qquad \text{Path constraints}$$

$$H_{eq}(p,q) = 0,\ H_{ineq}(p,q) \leq 0 \qquad \text{Point constraints}$$

where

$$\begin{aligned} v =& [\dot{x}^T, x^T, u^T, w^T, t]^T \\ q =& [\dot{x}(t_1)^T, x(t_1)^T, u(t_1)^T, w(t_1)^T, \ldots, \\ & \dot{x}(t_{n_{tp}})^T, x(t_{n_{tp}})^T, u(t_{n_{tp}})^T, w(t_{n_{tp}})^T]^T \end{aligned}$$

Optimization Mesh

- Divide the optimization interval into N_e intervals
- Introduce normalized element lengths $h_0, \ldots, h_{N_e-1}$ with

$$\sum_{i=0}^{N_e-1} h_i = 1$$

- Element junction points

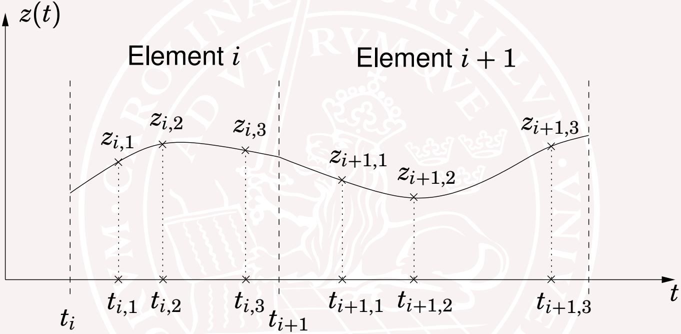
$$t_i = t_0 + (t_f - t_0) \sum_{k=0}^{i-1} h_k,\ i = 1..N_e - 1$$

- Radu collocation points $\tau_j \in (0..1]$, $j = 1..N_c$ in each element gives

$$t_{i,j} = t_0 + (t_f - t_0) \left(\sum_{k=0}^{i-1} h_k + \tau_j h_i \right),\ i = 0..N_e - 1,\ j = 1..N_c$$

Piecewise Polynomial Variables

Approximate variable profiles using piecewise polynomials



$$z(t) = \sum_{j=1}^{N_c} z_{i,j} L_j \left(\frac{t - t_{i-1}}{h_i} \right) \quad t \in [t_{i-1}, t_i], \quad h_i = t_{i+1} - t_i$$

L_j are interpolation polynomials

Lagrange Polynomials

Given N_c points, $\tau_1, \ldots, \tau_{N_c} \in [0,1]$, the corresponding Lagrange polynomials are given by

$$\begin{cases} L_j^{(N_c)}(\tau) = 1 \quad \text{if } N_c = 1 \\ L_j^{(N_c)}(\tau) = \prod_{k=1, k\neq j}^{N_c} \frac{\tau - \tau_k}{\tau_j - \tau_k} \quad \text{if } N_c \geq 2 \end{cases}$$

Property of Lagrange polynomials

- $L_j^{(N_c)}(\tau_k) = \delta_{j,k}$, i.e.,

$$L_j^{(N_c)}(\tau_k) = \begin{cases} 1, & \text{if} \quad j = k \\ 0, & \text{if} \quad j \neq k \end{cases}$$

It follows that

$$z(t_{i,j}) = \sum_{k=1}^{N_c} z_{i,k} L_k^{(N_c)} \left(\frac{t_{i,j} - t_{i-1}}{h_i} \right) = \sum_{k=1}^{N_c} z_{i,k} L_j^{(N_c)}(\tau_k) = z_{i,j}$$

Variable Approximation

- Variables approximated by Lagrange polynomials based on Radau points $\tau_1 \ldots \tau_{N_c}$ where $\tau_{N_c} = 1$
- State variables approximated by polynomials based on $N_c + 1$ points: add $\tau_0 = 0$.
- Controls and algebraic variables approximated by polynomials based on N_c points
- Variables in algebraic optimization problem
  - At initial point t_0: $\dot{x}_{0,0}$, $x_{0,0}$, $u_{0,0}$, $w_{0,0}$
  - At the collocation points: $\dot{x}_{i,j}$, $x_{i,j}$, $u_{i,j}$, $w_{i,j}$, $i = 0..N_e - 1$, $j = 1..N_c$,
  - At the element junction points t_i: $x_{i,0}$, $i = 1..N_e$
  - At the time points: $\dot{x}_i^p$, x_i^p, u_i^p, w_i^p, $i \in 1..N_{tp}$
- Number of variables in optimization vector $\bar{x}$: $N_{\bar{x}} = n_p + (2n_x + n_u + n_w)(n_e n_c + 1 + n_{tp}) n_x n_e + n_e$

Equality Constraints

- Initial equations

$$F_0(p, v_{0,0}) = 0,\ v_{0,0} = [\dot{x}_{0,0}^T, x_{0,0}^T, u_{0,0}^T, w_{0,0}^T, t_0]^T$$

- DAE dynamics at collocation points

$$F(p, v_{i,j}) = 0,\ i = 0..N_e - 1,\ j = 1..N_c$$

- Continuity of state profiles

$$x_{i,n_c} - x_{i+1,0} = 0,\ i = 0..N_e - 1$$

- Control variables at initial point (interpolation

$$u_{0,0} = \sum_{k=1}^{(N_c)} u_{0,k} L_k^{N_c}(0)$$

- Collocation equations

$$\dot{x}_{i,j} = \frac{1}{h_i(t_f - t_0)} \sum_{k=0}^{N_c} x_{i,k} \dot{L}_k^{(N_c+1)}(\tau_j),\ i = 0..N_e{-}1,\ j = 1..N_c$$

Equality Constraints cont'd

- Interpolation of variables at time points
  - $\dot{x}_l^p = \frac{1}{h_{i_l}(t_f-t_0)} \sum_{k=0}^{N_c} x_{i_l,k} \dot{L}_k^{(N_c+1)}(\tau_l^p),\ l = 1..n_{tp}$
  - $x_l^p = \sum_{k=0}^{N_c} x_{i_l,k} L_k^{(N_c+1)}(\tau_l^p),\ l = 1..n_{tp}$
  - $u_l^p = \sum_{k=1}^{N_c} u_{i_l,k} L_k^{(N_c)}(\tau_l^p),\ l = 1..n_{tp}$
  - $w_l^p = \sum_{k=1}^{N_c} w_{i_l,k} L_k^{(N_c)}(\tau_l^p),\ l = 1..n_{tp}$
- Total number of equality constraints resulting from discretization of dynamics: $2n_x + n_w + (n_x + n_w)N_eN_c + n_xN_e + n_u + n_xN_eN_c + (2n_x + n_u + n_w)n_{tp}$
- Degrees of freedom of algebraic optimization problem: $n_p + n_uN_eN_c$

A Non-Linear Program

Non-Linear program resulting from collocation

$$\min_{\bar{x}} f(\bar{x})$$

subject to

$$g(\bar{x}) \leq 0$$
$$h(\bar{x}) = 0$$

- g contains point and path inequality constraints C_{ineq} and H_{ineq}
- h contains point and path equality constraints C_{ineq} and H_{ineq} in addition to dynamic constraints

Collocation with Lagrange Polynomials – Properties

- Equivalent to an implicit Runge-Kutta one-step method
  - Large body of applicable theory from numerical analysis
- Good numerical stability properties
  - Applicable to stiff ($\approx$ numerically difficult) systems
- State and control constraints straightforward
- Handles unstable systems
- Accurate derivatives essential for convergence
- The NLP problem is usually very large but *sparse*, must be exploited for efficiency

What can go wrong?

- Convergence of gradient based methods relies on a twice continuous differentiable right-hand side ($\approx$ smooth $f(x, u)$)
- $\Rightarrow$ Discontinuities may cause problem
  - `if`-clauses (which introduce discontinuous)
    - Avoid, if possible (or use a method explicitly adressing discontinuities)
  - `abs`, `min` and `max` functions
    - Use $\max(x, y) = ((x - y)^2 + \epsilon^2)^{0.5}/2 + (x + y)/2$
  - Saturation
    - Use smooth approximation (can be constructed from smooth `min` and `max` approximations)
  - Lookup-tables
    - Use sufficiently smooth spline interpolations instead of linear interpolation
- Scaling problems (especially for simultaneous methods)
- Need a reasonable initial guess (use simulation)