Optimal Control
Lectures 19-20: Direct Solution Methods

Benoît Chachuat <benoit@mcmaster.ca>

McMaster University

Department of Chemical Engineering

Spring 2009

Optimal Control Formulation

We are concerned with numerical solution procedures for optimal control problems that comply with the following general formulation:

$$
\begin{aligned}
\text{Determine: } & \mathbf{u} \in \hat{\mathcal{C}}[t_0, t_f]^{n_u} \text{ and } \mathbf{v} \in \mathbb{R}^{n_v} \text{ that} \\
\text{minimize: } & \phi(\mathbf{x}(t_f), \mathbf{v}) \\
\text{subject to: } & \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}); \quad \mathbf{x}(t_0) = \mathbf{h}(\mathbf{v}) \\
& \boldsymbol{\psi}^{\mathrm{i}}(\mathbf{x}(t_f), \mathbf{v}) \leq \mathbf{0} \\
& \boldsymbol{\psi}^{\mathrm{e}}(\mathbf{x}(t_f), \mathbf{v}) = \mathbf{0} \\
& \boldsymbol{\kappa}^{\mathrm{i}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) \leq \mathbf{0}, \quad t_0 \leq t \leq t_f \\
& \boldsymbol{\kappa}^{\mathrm{e}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) = \mathbf{0}, \quad t_0 \leq t \leq t_f \\
& \mathbf{u}(t) \in [\mathbf{u}^L, \mathbf{u}^U], \qquad \mathbf{v} \in [\mathbf{v}^L, \mathbf{v}^U]
\end{aligned}
$$

Direct Methods of Optimal Control

Principle

Discretize the control problem, then apply NLP techniques to the resulting finite-dimensional optimization problem

- Studied extensively over the last 30 years
- Proved successful for many complex applications
- Take advantage of the power of state-of-the-art NLP solvers
- Can be applied to ODE, DAE and even PDAE models

Three Main Variants

1. Direct simultaneous approach
   - or orthogonal collocation, or full discretization
2. Direct sequential approach
   - or single-shooting or control vector parameterization (CVP)
3. Direct multiple-shooting approach

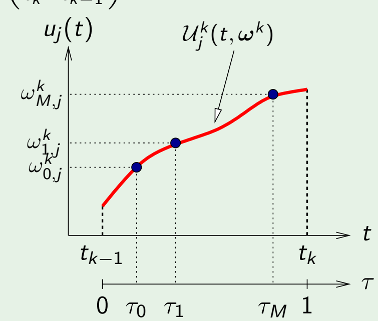
Control Parameterization

1. Subdivide the optimization horizon $[t_0, t_f]$ into $n_s \geq 1$ control stages, $t_0 < t_1 < t_2 < \cdots < t_{n_s} = t_f$
2. In each subinterval $[t_{k-1}, t_k]$, approximate $\mathbf{u}(t) = \boldsymbol{\mathcal{U}}^k(t, \boldsymbol{\omega}^k)$

Lagrange Interpolating Polynomials (Degree M)

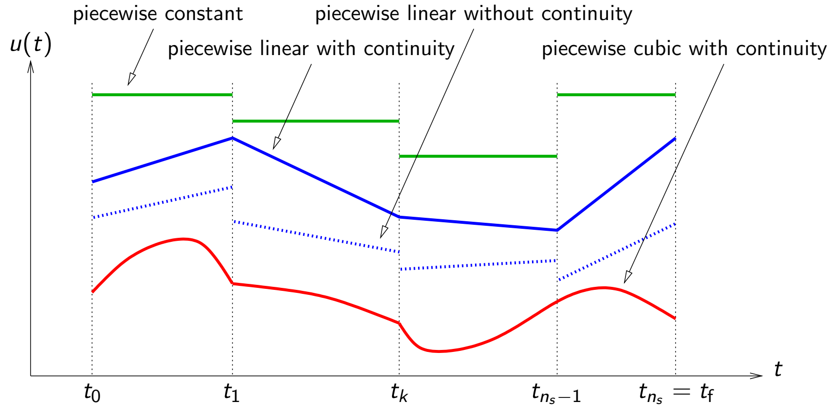
$$u_j(t) = \mathcal{U}_j^k(t, \boldsymbol{\omega}^k) \triangleq \sum_{i=0}^{M} \omega_{i,j}^k \phi_i^{(M)}\left(\frac{t - t_{k-1}}{t_k - t_{k-1}}\right), \quad t_{k-1} \leq t \leq t_k$$

$$\phi_i^{(M)}(\tau) \triangleq \begin{cases} 1, & \text{if } M = 0 \\ \displaystyle\prod_{\substack{q=0 \\ q \neq i}}^{M} \frac{\tau - \tau_q}{\tau_i - \tau_q}, & \text{if } M \geq 1 \end{cases}$$



with collocation points $0 \leq \tau_0 \leq \tau_1 < \cdots < \tau_{M-1} < \tau_M \leq 1$

Control Parameterization



- Control bounds can be enforced by bounding the control coefficients $\boldsymbol{\omega}^k$ in each subinterval
- Control continuity (or higher-order continuity) at stage times t_k can be enforced via linear inequality constraints between $\boldsymbol{\omega}^k$ and $\boldsymbol{\omega}^{k+1}$
- The choice of the collocation points τ_i has no effect on the solution

Direct Simultaneous Method

Principle

Transcription into a finite-dimensional NLP through discretization of both control and state variables

State Collocation

$$\mathbf{x}(t) = \boldsymbol{\mathcal{X}}(t, \boldsymbol{\xi}^k), \quad t_{k-1} \leq t \leq t_k, \; k = 1, \ldots, n_s$$

- Lagrange Polynomial Representation (Degree N)

$$\mathcal{X}_j(t, \boldsymbol{\xi}_j^k) \triangleq \sum_{i=0}^{N} \xi_{i,j}^k \phi_i^{(N)}\left(\frac{t - t_{k-1}}{t_k - t_{k-1}}\right), \quad t_{k-1} \leq t \leq t_k$$

$$\text{with:} \quad \phi_i^{(N)}(\tau_q) = \delta_{i,q}, \quad q = 0, \ldots, N$$

Direct Simultaneous Method

Principle

Transcription into a finite-dimensional NLP through discretization of both control and state variables

State Collocation

$$\mathbf{x}(t) = \boldsymbol{\mathcal{X}}(t, \boldsymbol{\xi}^k), \quad t_{k-1} \leq t \leq t_k, \; k = 1, \ldots, n_s$$

- Monomial Basis Representation (Degree N)

$$\mathcal{X}_j(t, \boldsymbol{\xi}_j^k) \triangleq \xi_{0,j}^k + (t_k - t_{k-1}) \sum_{i=1}^{N} \xi_{i,j}^k \Omega_i^{(N)}\left(\frac{t - t_{k-1}}{t_k - t_{k-1}}\right), \quad t_{k-1} \leq t \leq t_k$$

$$\text{with:} \quad \Omega_i^{(N)}(0) = 0$$
$$\dot{\Omega}_i^{(N)}(\tau_q) = \delta_{i,q}, \quad q = 1, \ldots, N$$

Direct Simultaneous Method (cont'd)

Original Optimal Control Problem

Determine: $\mathbf{u} \in \hat{\mathcal{C}}[t_0, t_f]^{n_u}$ and $\mathbf{v} \in \mathbb{R}^{n_v}$ that

minimize: $\phi(\mathbf{x}(t_f), \mathbf{v})$

subject to:
$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}); \quad \mathbf{x}(t_0) = \mathbf{h}(\mathbf{v})$$
$$\boldsymbol{\psi}^{\mathrm{i}}(\mathbf{x}(t_f), \mathbf{v}) \leq \mathbf{0}$$
$$\boldsymbol{\psi}^{\mathrm{e}}(\mathbf{x}(t_f), \mathbf{v}) = \mathbf{0}$$
$$\boldsymbol{\kappa}^{\mathrm{i}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) \leq \mathbf{0}, \quad t_0 \leq t \leq t_f$$
$$\boldsymbol{\kappa}^{\mathrm{e}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) = \mathbf{0}, \quad t_0 \leq t \leq t_f$$
$$\mathbf{u}(t) \in [\mathbf{u}^L, \mathbf{u}^U], \qquad \mathbf{v} \in [\mathbf{v}^L, \mathbf{v}^U]$$

Direct Simultaneous Method (cont'd)

Fully-Discretized Optimization Problem

Determine: $\boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^{n_s} \in \mathbb{R}^{n_u M}, \boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^{n_s} \in \mathbb{R}^{n_x N}$ and $\mathbf{v} \in \mathbb{R}^{n_v}$ that

minimize: $\phi(\mathcal{X}^{n_s}(t_{n_s}, \boldsymbol{\xi}^{n_s}), \mathbf{v})$

subject to:
$$\mathcal{X}_t^k(t_{k,q}, \boldsymbol{\xi}^k) = \mathbf{f}(t_{k,q}, \mathcal{X}^k(t_{k,q}, \boldsymbol{\xi}^k), \mathcal{U}^k(t_{k,q}, \boldsymbol{\omega}^k), \mathbf{v}), \quad \forall k, q$$
$$\mathcal{X}^1(t_0, \boldsymbol{\xi}^1) = \mathbf{h}(\mathbf{v}); \quad \mathcal{X}^{k+1}(t_k, \boldsymbol{\xi}^{k+1}) = \mathcal{X}^k(t_k, \boldsymbol{\xi}^k), \quad \forall k$$
$$\boldsymbol{\psi}^{\mathrm{i}}(\mathcal{X}^{n_s}(t_{n_s}, \boldsymbol{\xi}^{n_s}), \mathbf{v}) \leq 0$$
$$\boldsymbol{\psi}^{\mathrm{e}}(\mathcal{X}^{n_s}(t_{n_s}, \boldsymbol{\xi}^{n_s}), \mathbf{v}) = 0$$
$$\boldsymbol{\kappa}^{\mathrm{i}}(t_{k,q}, \mathcal{X}^k(t_{k,q}, \boldsymbol{\xi}^k), \mathcal{U}^k(t_{k,q}, \boldsymbol{\omega}^k), \mathbf{v}) \leq 0, \quad \forall k, q$$
$$\boldsymbol{\kappa}^{\mathrm{e}}(t_{k,q}, \mathcal{X}^k(t_{k,q}, \boldsymbol{\xi}^k), \mathcal{U}^k(t_{k,q}, \boldsymbol{\omega}^k), \mathbf{v}) = 0, \quad \forall k, q$$
$$\boldsymbol{\xi}^k \in [\boldsymbol{\xi}^L, \boldsymbol{\xi}^U], \quad \boldsymbol{\omega} \in [\boldsymbol{\omega}^L, \boldsymbol{\omega}^U], \quad \mathbf{v} \in [\mathbf{v}^L, \mathbf{v}^U]$$

with $t_{k,q} \triangleq t_{k-1} + \tau_q(t_k - t_{k-1})$, $k = 1, \ldots, n_s$, $q = 1, \ldots, N$

Direct Simultaneous Method (cont'd)

Class Exercise: Discretize the following optimal control problem into an NLP via the full discretization approach:

$$\min_{u(t)} \int_0^1 \frac{1}{2}[u(t)]^2 \, dt$$
$$\text{s.t.} \quad \dot{x}(t) = u(t) - x(t); \quad x(0) = 1$$
$$x(1) = 0$$

- Consider a single control stage, $n_s = 1$
- Approximate the state and control profiles using affine functions, $M = N = 1$

Direct Simultaneous Method (cont'd)

Pros and Cons

- very large-scale NLP problems in the variables $\mathbf{p}^{\mathsf{T}} = (\boldsymbol{\xi}^{1\mathsf{T}}, \ldots, \boldsymbol{\xi}^{n_s \mathsf{T}}, \boldsymbol{\omega}^{1\mathsf{T}}, \ldots, \boldsymbol{\omega}^{n_s \mathsf{T}}, \mathbf{v}^{\mathsf{T}})$
- The numbers of time stages and collocation points as well as the position of the collocation points must be chosen *a priori*
- The stage times can be optimized as part of the decision vector $\mathbf{p}$ too
- Infeasible path method:
  - The ODEs are satisfied at the converged solution of the NLP only
  - But, this saves computational effort and allows unstable systems
- Path constraints are easily accommodated by enforcing inequality constraint at the collocation points

Class Exercise: Give sufficient conditions for existence of an optimal solution to the fully-discretized optimization problem

Direct Sequential Method

Principle

Transcription into a finite-dimensional NLP through discretization of the control variables only, while the ODEs are embedded in the NLP problem

Original Optimal Control Problem

Determine: $\mathbf{u} \in \hat{\mathcal{C}}[t_0, t_f]^{n_u}$ and $\mathbf{v} \in \mathbb{R}^{n_v}$ that

minimize: $\phi(\mathbf{x}(t_f), \mathbf{v})$

subject to:
$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}); \quad \mathbf{x}(t_0) = \mathbf{h}(\mathbf{v})$$
$$\boldsymbol{\psi}^{\mathrm{i}}(\mathbf{x}(t_f), \mathbf{v}) \leq \mathbf{0}$$
$$\boldsymbol{\psi}^{\mathrm{e}}(\mathbf{x}(t_f), \mathbf{v}) = \mathbf{0}$$
$$\boldsymbol{\kappa}^{\mathrm{i}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) \leq \mathbf{0}, \quad t_0 \leq t \leq t_f$$
$$\boldsymbol{\kappa}^{\mathrm{e}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) = \mathbf{0}, \quad t_0 \leq t \leq t_f$$
$$\mathbf{u}(t) \in [\mathbf{u}^L, \mathbf{u}^U], \quad \mathbf{v} \in [\mathbf{v}^L, \mathbf{v}^U]$$

Direct Sequential Method (cont'd)

Partially-Discretized Optimization Problem

Determine: $\omega^1, \ldots, \omega^{n_s} \in \mathbb{R}^{n_u M}$ and $\mathbf{v} \in \mathbb{R}^{n_v}$ that

minimize: $\phi(\mathbf{x}(t_{n_s}), \mathbf{v})$

subject to:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathcal{U}^k(t, \omega^k), \mathbf{v}); \quad \mathbf{x}(t_0) = \mathbf{h}(\mathbf{v})$$

$$\psi^{\mathrm{i}}(\mathbf{x}(t_{n_s}), \mathbf{v}) \leq \mathbf{0}$$

$$\psi^{\mathrm{e}}(\mathbf{x}(t_{n_s}), \mathbf{v}) = \mathbf{0}$$

$$\kappa^{\mathrm{i}}(t, \mathbf{x}(t), \mathcal{U}^k(t, \omega^k), \mathbf{v}) \leq \mathbf{0}, \quad t_{k-1} \leq t \leq t_k, \ \forall k$$

$$\kappa^{\mathrm{e}}(t, \mathbf{x}(t), \mathcal{U}^k(t, \omega^k), \mathbf{v}) = \mathbf{0}, \quad t_{k-1} \leq t \leq t_k, \ \forall k$$

$$\omega(t) \in [\omega^L, \omega^U], \qquad \mathbf{v} \in [\mathbf{v}^L, \mathbf{v}^U]$$

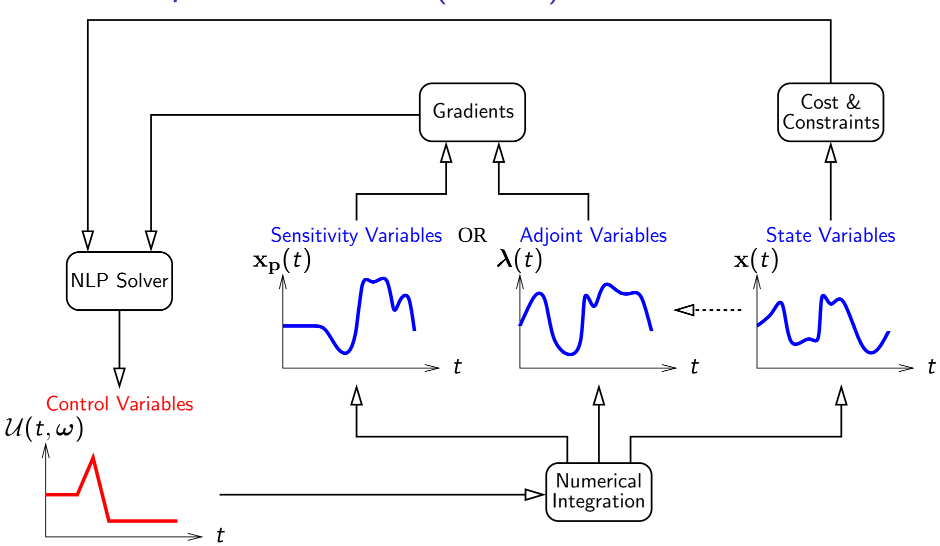
Direct Sequential Method (cont'd)

Class Exercise: Discretize the following optimal control problem into an NLP via the sequential approach:

$$\begin{aligned} \min_{u(t)} \quad & \int_0^1 \frac{1}{2}[u(t)]^2 \, dt \\ \text{s.t.} \quad & \dot{x}(t) = u(t) - x(t); \quad x(0) = 1 \\ & x(1) = 0 \end{aligned}$$

- Consider a single control stage, $n_s = 1$
- Approximate the control profile using affine functions, $M = 1$

Direct Sequential Method (cont'd)



Issue 1: How to calculate the cost and constraint values and derivatives?

Direct Sequential Method (cont'd)

Issue 2: How to handle path constraints?

Reformulation as integral constraints:

$$\mathcal{K}_j^{\mathrm{e}}(\mathbf{p}) := \sum_{k=1}^{n_s} \int_{t_{k-1}}^{t_k} \left(\kappa_j^{\mathrm{i}}(t, \mathbf{x}(t), \mathcal{U}^k(t, \omega^k), \mathbf{v}) \right)^2 \, dt$$

$$\mathcal{K}_j^{\mathrm{i}}(\mathbf{p}) := \sum_{k=1}^{n_s} \int_{t_{k-1}}^{t_k} \max \left\{ 0; \kappa_j^{\mathrm{i}}(t, \mathbf{x}(t), \mathcal{U}^k(t, \omega^k), \mathbf{v}) \right\}^2 \, dt$$

But, relaxation needed to ensure regular constraints:

$$\mathcal{K}_j(\mathbf{p}) \leq \epsilon$$

with $\epsilon > 0$ a small nonnegative constant

Direct Sequential Method (cont'd)

Issue 2: How to handle path constraints?

Discretization as interior point constraints:

$$\kappa_j^{\mathrm{i}}\left(t_{k,q}, \mathbf{x}(t_{k,q}), \boldsymbol{\mathcal{U}}^k(t_{k,q}, \boldsymbol{\omega}^k), \mathbf{v}\right) \leq 0$$

$$\kappa_j^{\mathrm{e}}\left(t_{k,q}, \mathbf{x}(t_{k,q}), \boldsymbol{\mathcal{U}}^k(t_{k,q}, \boldsymbol{\omega}^k), \mathbf{v}\right) = 0$$

at a given set of points $t_{k,q} \in [t_{k-1}, t_k]$ in each stage $k = 1, \ldots, n_s$

The combination of reformulation and discretization approaches is recommended!

Direct Sequential Method (cont'd)

Pros and Cons

- Relatively small-scale NLP problems in the variables $\mathbf{p}^{\mathsf{T}} = (\boldsymbol{\omega}^{1\,\mathsf{T}}, \ldots, \boldsymbol{\omega}^{n_s\,\mathsf{T}}, \mathbf{v}^{\mathsf{T}})$
- Stage times can be optimized as part of the decision vector $\mathbf{p}$ too
- The accuracy of the state variables is enforced via the error-control mechanism of state-of-the-art numerical solvers
- Feasible path method:
  - The ODEs are satisfied at each iteration of the NLP algorithm
  - But, this is computationally demanding and handles mildly unstable systems only

Class Exercise: Give sufficient conditions for existence of an optimal solution to the partially-discretized optimization problem

Scalar Dynamic Optimization Example

Consider the dynamic optimization problem to:

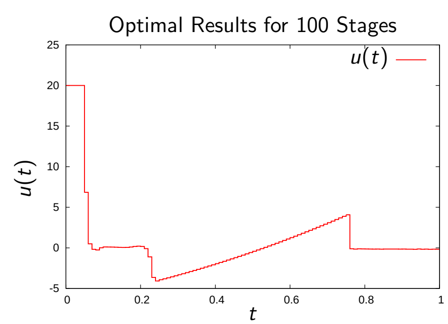
$$\begin{aligned}
\textbf{minimize:} \quad & \mathcal{J}(u) := \int_0^1 \left([x_1(t)]^2 + [x_2(t)]^2\right)\, \mathrm{d}t \\
\textbf{subject to:} \quad & \dot{x}_1(t) = x_2(t); \quad x_1(0) = 0 \\
& \dot{x}_2(t) = -x_2(t) + u(t); \quad x_2(0) = -1 \\
& x_2(t) + 0.5 - 8[t - 0.5]^2 \leq 0, \forall t \\
& -20 \leq u(t) \leq 20, \forall t
\end{aligned}$$

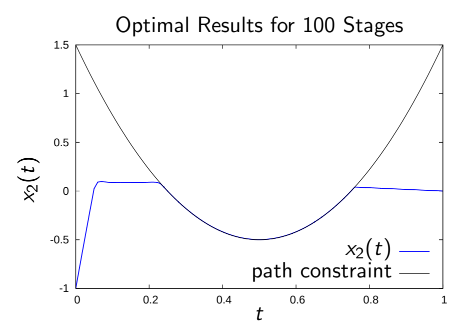
Solution Approach Used

- Direct sequential approach with piecewise constant controls
- Nonlinear program solved with SQP methods
- Inequality path constraint reformulated as:

$$\int_0^1 \max\{0; x_2(t) + 0.5 - 8[t - 0.5]^2\}^2 \mathrm{d}t \leq \epsilon, \quad \text{with } \epsilon = 10^{-6}$$

Scalar Dynamic Optimization Example (cont'd)





Optimal Cost

n_s	10	20	40	100
$\mathcal{J}(u^\star)$	1.13080×10^{-1}	0.97320×10^{-1}	0.96942×10^{-1}	0.96893×10^{-1}

Multiple-Shooting Method

Principle

Transcription into a finite-dimensional NLP through discretization of the control variables only, with state discontinuities allowed at stage times

Extra Decision Variables and Constraints

- New decision variables $\boldsymbol{\xi}_0^k \in \mathbb{R}^{n_x}$, $k = 2, \ldots, n_s$, such that

$$\dot{\mathbf{x}}^k(t) = \mathbf{f}(t, \mathbf{x}^k(t), \mathcal{U}^k(t, \boldsymbol{\omega}^k), \mathbf{v}), \quad t_{k-1} \leq t \leq t_k$$

$$\text{with:} \quad \mathbf{x}^k(t_{k-1}) = \begin{cases} \mathbf{h}(\mathbf{v}) & \text{if } k = 1, \\ \boldsymbol{\xi}_0^k & \text{otherwise.} \end{cases}$$

- New (nonlinear) equality constraints:

$$\mathbf{x}^k(t_k) - \boldsymbol{\xi}_0^{k+1} = \mathbf{0}, \quad k = 1, \ldots, n_s - 1$$

Multiple-Shooting Method (cont'd)

Original Optimal Control Problem

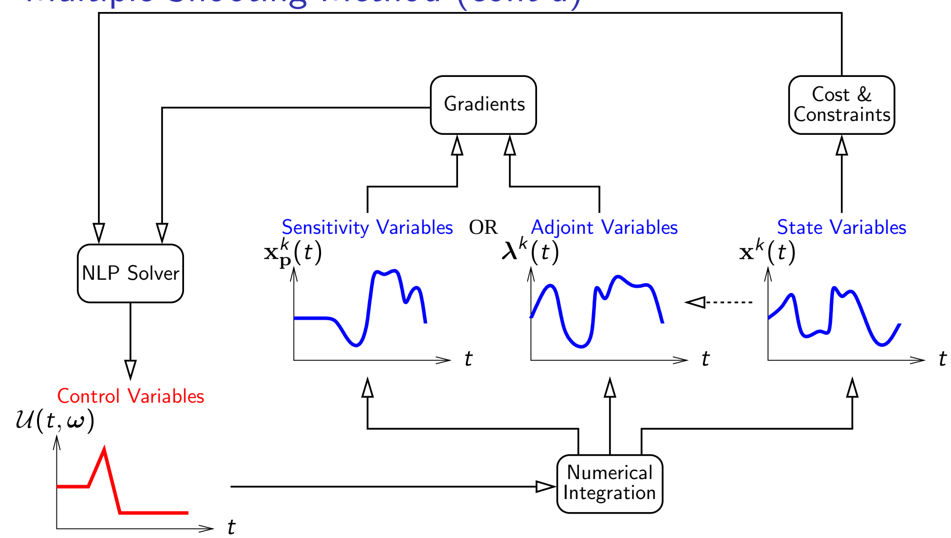
$$\begin{aligned}
\text{Determine:} \quad & \mathbf{u} \in \hat{\mathcal{C}}[t_0, t_f]^{n_u} \text{ and } \mathbf{v} \in \mathbb{R}^{n_v} \text{ that} \\
\text{minimize:} \quad & \phi(\mathbf{x}(t_{n_s}), \mathbf{v}) \\
\text{subject to:} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}); \quad \mathbf{x}(t_0) = \mathbf{h}(\mathbf{v}) \\
& \boldsymbol{\psi}^{\mathrm{i}}(\mathbf{x}(t_{n_s}), \mathbf{v}) \leq \mathbf{0} \\
& \boldsymbol{\psi}^{\mathrm{e}}(\mathbf{x}(t_{n_s}), \mathbf{v}) = \mathbf{0} \\
& \boldsymbol{\kappa}^{\mathrm{i}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) \leq \mathbf{0}, \quad t_0 \leq t \leq t_f \\
& \boldsymbol{\kappa}^{\mathrm{e}}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{v}) = \mathbf{0}, \quad t_0 \leq t \leq t_f \\
& \mathbf{u}(t) \in [\mathbf{u}^L, \mathbf{u}^U], \quad \mathbf{v} \in [\mathbf{v}^L, \mathbf{v}^U]
\end{aligned}$$

Multiple-Shooting Method (cont'd)

Partially-Discretized Optimization Problem

$$\begin{aligned}
\text{Determine:} \quad & \boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^{n_s} \in \mathbb{R}^{n_u M}, \boldsymbol{\xi}_0^1, \ldots, \boldsymbol{\xi}_0^{n_s} \in \mathbb{R}^{n_x} \text{ and } \mathbf{v} \in \mathbb{R}^{n_v} \text{ that} \\
\text{minimize:} \quad & \phi(\mathbf{x}(t_f), \mathbf{v}) \\
\text{subject to:} \quad & \dot{\mathbf{x}}^k(t) = \mathbf{f}(t, \mathbf{x}^k(t), \mathcal{U}^k(t, \boldsymbol{\omega}^k), \mathbf{v}), \quad t_{k-1} \leq t \leq t_k, \ \forall k \\
& \mathbf{x}^1(t_0) = \mathbf{h}(\mathbf{v}); \quad \mathbf{x}^k(t_{k-1}) = \boldsymbol{\xi}_0^k, \quad k = 2, \ldots, n_s \\
& \boldsymbol{\psi}^{\mathrm{i}}(\mathbf{x}(t_f), \mathbf{v}) \leq \mathbf{0} \\
& \boldsymbol{\psi}^{\mathrm{e}}(\mathbf{x}(t_f), \mathbf{v}) = \mathbf{0} \\
& \boldsymbol{\kappa}^{\mathrm{i}}(t, \mathbf{x}(t), \mathcal{U}^k(t, \boldsymbol{\omega}^k), \mathbf{v}) \leq \mathbf{0}, \quad t_{k-1} \leq t \leq t_k, \ \forall k \\
& \boldsymbol{\kappa}^{\mathrm{e}}(t, \mathbf{x}(t), \mathcal{U}^k(t, \boldsymbol{\omega}^k), \mathbf{v}) = \mathbf{0}, \quad t_{k-1} \leq t \leq t_k, \ \forall k \\
& \boldsymbol{\xi}_0^k \in [\boldsymbol{\xi}_0^L, \boldsymbol{\xi}_0^U], \quad \boldsymbol{\omega}(t) \in [\boldsymbol{\omega}^L, \boldsymbol{\omega}^U], \quad \mathbf{v} \in [\mathbf{v}^L, \mathbf{v}^U]
\end{aligned}$$

Multiple-Shooting Method (cont'd)



Exact same procedure as the sequential approach, except that multiple independent sets of differential equations

Multiple-Shooting Method

Pros and Cons

- Lies between the sequential and simultaneous approaches: **sequential infeasible-path method**
- Shares many attractive features of the sequential and simultaneous approaches:
  - accurate solution of the differential equations
  - ability to deal with unstable systems safely
- SQP solvers exploiting the special structure of the resulting NLP problem can/must be devised
- Multiple shooting lends itself naturally to parallelization

Class Exercise: Give sufficient conditions for existence of an optimal solution to the partially-discretized optimization problem

Handling Functionals with State Variables Participating

Consider the functional in Mayer form:

$$\mathcal{F}(\mathbf{p}) \stackrel{\Delta}{=} \phi(\mathbf{x}(t_f), \mathbf{p})$$

subject to the parametric initial value problem:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{p}), \quad t_0 \le t \le t_f; \qquad \mathbf{x}(t_0) = \mathbf{h}(\mathbf{p})$$

Key Issues in Sequential Approaches:

Given that a unique solution $\mathbf{x}(t; \bar{\mathbf{p}})$ to the ODEs exists for given $\bar{\mathbf{p}} \in P$,

1. Calculate the functional value, $\mathcal{F}(\bar{\mathbf{p}})$
   - Use state-of-the-art methods for differential equations
2. Calculate the functional derivatives, $\mathcal{F}_{p_j}(\bar{\mathbf{p}})$, w.r.t. $p_1, \ldots, p_{n_p}$
   - 3 possible procedures: finite differences, forward sensitivity analysis, adjoint sensitivity analysis

Functional Derivatives

Finite Differences Approach:

$$\mathcal{F}_{p_i}(\mathbf{p}) \approx \frac{\phi(\mathbf{x}(t_f), p_1, \ldots, p_i + \delta p_i, \ldots, p_{n_p}) - \phi(\mathbf{x}(t_f), \mathbf{p})}{\delta p_i}$$

Procedure for Finite Differences Approach

- Initial Step:
  Solve ODEs for the actual parameter values $\bar{\mathbf{p}}$;
  Calculate the value of $\mathcal{F}(\bar{\mathbf{p}})$
- Loop: $j = 1, \ldots, n_p$
  Set $\tilde{p}_i := \bar{p}_i$, $i \ne j$; $\tilde{p}_j := \bar{p}_j + \delta p_j$;
  Solve ODEs for the perturbed parameter values $\tilde{\mathbf{p}}$
  Calculate the value of $\mathcal{F}(\tilde{\mathbf{p}})$, and $\mathcal{F}_{p_j}(\bar{\mathbf{p}}) \approx \dfrac{\mathcal{F}(\tilde{\mathbf{p}}) - \mathcal{F}(\bar{\mathbf{p}})}{\delta p_j}$
- End Loop

Trade-off: Choose δp_j small enough, but not too small!

Functional Derivatives (cont'd)

State Sensitivities: Under which conditions is $\mathbf{x}(t; \cdot)$ continuously differentiable w.r.t. $p_1, \ldots, p_{n_p}$, at a point $\bar{\mathbf{p}} \in P$, for $t_0 \le t \le t_f$?

1. A unique solution $\mathbf{x}(t; \bar{\mathbf{p}})$ to the ODEs exists on $[t_0, t_f]$
2. $\mathbf{f}$ is continuously differentiable w.r.t. $\mathbf{x}$ and $\mathbf{p}$, and piecewise continuous w.r.t. t
3. $\mathbf{h}$ is continuously differentiable w.r.t. $\mathbf{p}$

Sensitivity Equations:

The state sensitivities, $\mathbf{x}_{p_j}(t) \stackrel{\Delta}{=} \frac{\partial \mathbf{x}(t)}{\partial p_j}$, satisfy the ODEs:

$$\dot{\mathbf{x}}_{p_j}(t) = \mathbf{f}_{\mathbf{x}}(t, \mathbf{x}(t), \bar{\mathbf{p}})\, \mathbf{x}_{p_j}(t) + \mathbf{f}_{p_j}(t, \mathbf{x}(t), \bar{\mathbf{p}})$$

with the initial conditions $\mathbf{x}_{p_j}(t_0) = \mathbf{h}_{p_j}(\bar{\mathbf{p}})$

- One sensitivity equation for each parameter $p_1, \ldots, p_{n_p}$!
- **Functional derivative:** $\mathcal{F}_{p_j}(\bar{\mathbf{p}}) = \phi_{\mathbf{x}}(\mathbf{x}(t_f), \bar{\mathbf{p}})\mathbf{x}_{p_j}(t_f) + \phi_{p_j}(\mathbf{x}(t_f), \bar{\mathbf{p}})$

Functional Derivatives (cont'd)

Procedure for Forward Sensitivity Approach

- State and Sensitivity Numerical Integration: $t_0 \to t_f$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \overline{\mathbf{p}}); \qquad \mathbf{x}(t_0) = \mathbf{h}(\overline{\mathbf{p}})$$
$$\dot{\mathbf{x}}_{p_1}(t) = \mathbf{f}_{\mathbf{x}}(t, \mathbf{x}(t), \overline{\mathbf{p}})\, \mathbf{x}_{p_1}(t) + \mathbf{f}_{p_1}(t, \mathbf{x}(t), \overline{\mathbf{p}}); \qquad \mathbf{x}_{p_1}(t_0) = \mathbf{h}_{p_1}(\overline{\mathbf{p}})$$
$$\vdots \qquad\qquad \vdots$$
$$\dot{\mathbf{x}}_{p_{n_p}}(t) = \mathbf{f}_{\mathbf{x}}(t, \mathbf{x}(t), \overline{\mathbf{p}})\, \mathbf{x}_{p_{n_p}}(t) + \mathbf{f}_{p_{n_p}}(t, \mathbf{x}(t), \overline{\mathbf{p}}); \qquad \mathbf{x}_{p_{n_p}}(t_0) = \mathbf{h}_{p_{n_p}}(\overline{\mathbf{p}})$$

- Function and Gradient Evaluation:

$$\mathcal{F}(\mathbf{p}) = \phi(\mathbf{x}(t_f), \overline{\mathbf{p}})$$
$$\mathcal{F}_{p_1}(\overline{\mathbf{p}}) = \phi_{\mathbf{x}}(\mathbf{x}(t_f), \overline{\mathbf{p}})\mathbf{x}_{p_1}(t_f) + \phi_{p_1}(\mathbf{x}(t_f), \overline{\mathbf{p}})$$
$$\vdots$$
$$\mathcal{F}_{p_{n_p}}(\overline{\mathbf{p}}) = \phi_{\mathbf{x}}(\mathbf{x}(t_f), \overline{\mathbf{p}})\mathbf{x}_{p_{n_p}}(t_f) + \phi_{p_{n_p}}(\mathbf{x}(t_f), \overline{\mathbf{p}})$$

Functional Derivatives (cont'd)

Forward Sensitivity Approach:

$$\mathcal{F}_{p_i}(\mathbf{p}) = \phi_{\mathbf{x}}(\mathbf{x}(t_f), \mathbf{p})\, \mathbf{x}_{p_i}(t_f) + \phi_{p_i}(\mathbf{x}(t_f), \mathbf{p})$$
with: $\dot{\mathbf{x}}_{p_i}(t) = \mathbf{f}_{\mathbf{x}}(t, \mathbf{x}(t), \mathbf{p})\, \mathbf{x}_{p_i}(t) + \mathbf{f}_{p_i}(t, \mathbf{x}(t), \mathbf{p}); \quad \mathbf{x}_{p_i}(t_0) = \mathbf{h}_{p_i}(\mathbf{p})$

- Number of state & state sensitivity equations: $(n_x + 1) \times n_p$
- But, independent of the number of functionals, $n_{\mathcal{F}}$

Adjoint (Reverse) Sensitivity Approach:

$$\mathcal{F}_{p_i}(\mathbf{p}) = \phi_{p_i}(\mathbf{x}(t_f), \mathbf{p}) + \boldsymbol{\lambda}(t_0)^{\mathsf{T}} \mathbf{h}_{p_i}(\mathbf{p}) + \int_{t_0}^{t_f} \mathbf{f}_{p_i}(t, \mathbf{x}(t), \mathbf{p})^{\mathsf{T}} \boldsymbol{\lambda}(t) \mathrm{d}t$$
with: $\dot{\boldsymbol{\lambda}}(t) = -\mathbf{f}_{\mathbf{x}}(t, \mathbf{x}(t), \mathbf{p})^{\mathsf{T}} \boldsymbol{\lambda}(t); \quad \boldsymbol{\lambda}(t_f) = \phi_{\mathbf{x}}(\mathbf{x}(t_f), \mathbf{p})$

- Number of state & adjoint equations: $2n_x \times n_{\mathcal{F}}$
- But, independent of the number of parameters

Functional Derivatives (cont'd)

Procedure for Adjoint Sensitivity Approach

- State Numerical Integration: $t_0 \to t_f$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \overline{\mathbf{p}}); \qquad \mathbf{x}(t_0) = \mathbf{h}(\overline{\mathbf{p}});$$

  Store state values $\mathbf{x}(t)$ at mesh points, $t_0 < t_1 < \cdots < t_M = t_f$

- Adjoint Numerical Integration: $t_f \to t_0$

$$\dot{\boldsymbol{\lambda}}(t) = -\mathbf{f}_{\mathbf{x}}(t, \mathbf{x}(t), \overline{\mathbf{p}})^{\mathsf{T}} \boldsymbol{\lambda}(t); \qquad \boldsymbol{\lambda}(t_f) = \phi_{\mathbf{x}}(\mathbf{x}(t_f), \overline{\mathbf{p}})^{\mathsf{T}}$$
$$\dot{q}_1(t) = -\mathbf{f}_{p_1}(t, \mathbf{x}(t), \overline{\mathbf{p}})^{\mathsf{T}} \boldsymbol{\lambda}(t); \qquad q_1(t_f) = 0$$
$$\vdots$$
$$\dot{q}_{n_p}(t) = -\mathbf{f}_{p_{n_p}}(t, \mathbf{x}(t), \overline{\mathbf{p}})^{\mathsf{T}} \boldsymbol{\lambda}(t); \qquad q_{n_p}(t_f) = 0;$$

  Need to interpolate state values $\mathbf{x}(t)$ between mesh points

Functional Derivatives (cont'd)

Procedure for Adjoint Sensitivity Approach (cont'd)

- Function and Gradient Evaluation:

$$\mathcal{F}(\mathbf{p}) = \phi(\mathbf{x}(t_f), \overline{\mathbf{p}})$$
$$\mathcal{F}_{p_1}(\overline{\mathbf{p}}) = \phi_{p_1}(\mathbf{x}(t_f), \overline{\mathbf{p}}) + \boldsymbol{\lambda}(t_0)^{\mathsf{T}} \mathbf{h}_{p_1}(\overline{\mathbf{p}}) + q_1(t_0)$$
$$\vdots$$
$$\mathcal{F}_{p_{n_p}}(\overline{\mathbf{p}}) = \phi_{p_{n_p}}(\mathbf{x}(t_f), \overline{\mathbf{p}}) + \boldsymbol{\lambda}(t_0)^{\mathsf{T}} \mathbf{h}_{p_{n_p}}(\overline{\mathbf{p}}) + q_{n_p}(t_0)$$

Functional Derivatives (cont'd)

Class Exercise: Consider the functional

$$\mathcal{F}(p_1, p_2) = x(1)$$

$$\text{with:} \quad \dot{x}(t) = -x(t) + p_1; \quad x(0) = p_2$$

1. Calculate the value of $\mathcal{F}$ by first solving the ODE analytically; then, calculate the derivatives $\mathcal{F}_{p_1}$ and $\mathcal{F}_{p_2}$
2. Derive the sensitivity equations and their respective initial conditions, then find a solution to these equations; apply the forward sensitivity formula to calculate $\mathcal{F}_{p_1}$ and $\mathcal{F}_{p_2}$
3. Derive the adjoint equation and its terminal condition, then find a solution to this equation; apply the adjoint sensitivity formula to calculate $\mathcal{F}_{p_1}$ and $\mathcal{F}_{p_2}$

Existence of an Optimal Solution (Finite Dimension)

Recalls: Weierstrass' Theorem

Let $P \subset \mathbb{R}^n$ be a nonempty, compact set, and let $\phi : P \to \mathbb{R}$ be continuous on P. Then, the problem $\min\{\phi(p) : p \in P\}$ attains its minimum, that is, there exists a minimizing solution to this problem.

Why closedness of P? Why continuity of ϕ? Why boundedness of P?