LPE Reading Group
30 Oct 01

Today: 1) Gradient method for Optim. Control Problems
2) Shooting (Forw., Back, Multiple)
3) RIOTS
4) Next 2 Lectures

Green
Red Erase
Blue
Black

$$\min_x L(x)$$

$$\frac{\partial L}{\partial f}^T = -\lambda \quad \text{s.t. } f(x) \; (= 0)$$

$$\min_{x,\lambda} L(x) + \lambda^T f(x)$$

$$\frac{\partial}{\partial x}, \frac{\partial}{\partial \lambda}$$

$$\lambda = -[f_x^{-1} L_x]^T$$

$$J^* = \min_{u(\cdot)} \left\{ \int_0^T L(x,u)\,dt + V(x(T)) \right\}$$

$$\delta J^* = (\quad) \delta u \quad \text{s.t. } \dot{x} = f(x,u)$$

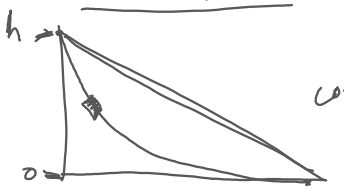
$$0 \qquad lb \le \chi(x(t)) \le ub \qquad t \in [0,T]$$

$$\lambda^T(t)[f - \dot{x}]$$

$$\dot{\lambda}^T = -f_x^{-1}\lambda - L_x$$

P.T) Gradient

Brachiostone



$H_u = 0$

- guess $u^0(t)$, $t \in [0, t_f]$
- compute $x^0(t)$ via $\dot{x}^0 = f(x^0(t), u^0(t))$ $x(t_0^0) = x_0$ given
- compute $\lambda^0(t)$ → $\dot{\lambda} = -f_x^T \lambda - L_x$ → ~~$\lambda(t_f)$~~ $\lambda(t_f) = \frac{\partial \phi}{\partial x(t_f)}^T$
- compute $H_u = (L_u + \lambda^T f_u)(t)$
- ~~to~~ set $\delta u = -\varepsilon H_u^T(t)$
- repeat until update → $u' = u^0 + \delta u$, $u^0 \leftarrow u'$

$\delta J = -\varepsilon \int_{t_0}^{t_f} (H_u^T H_u)\, dt$ small

$\underline{u}^*(x,\lambda) \leftarrow \frac{\partial H}{\partial u} = 0 \Rightarrow$

$\dot{x} = g_1(x,\lambda,t)$, $x(t_0)$ given

$\dot{\lambda} = g_2(x,\lambda,t)$, $\lambda(t_f) = \left(\frac{\partial \phi}{\partial x(t_f)}\right)^T$

Find: $x^*(t)$, $\lambda^*(t)$

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

$x(t;x_0) = x_0 + \int_{t_0}^{t} f(x,u)\,d\tau$

Forw. shooting a) guess $\lambda(t_0)$

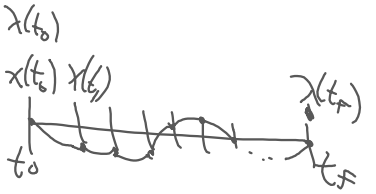
b) integ. forward

⋮

Back Shoot. a) $x(t_f)$

b) integ. back.

Multiple Shoot. →



RIOTS Recursive Integration Optimal Trajectory Solver

problem

$$\min \left\{ \phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x,u,t)\,dt \right\}$$

s.t. $\dot{x} = f(x,u,t)$, $x(t_0)$ given $\lambda(t)$

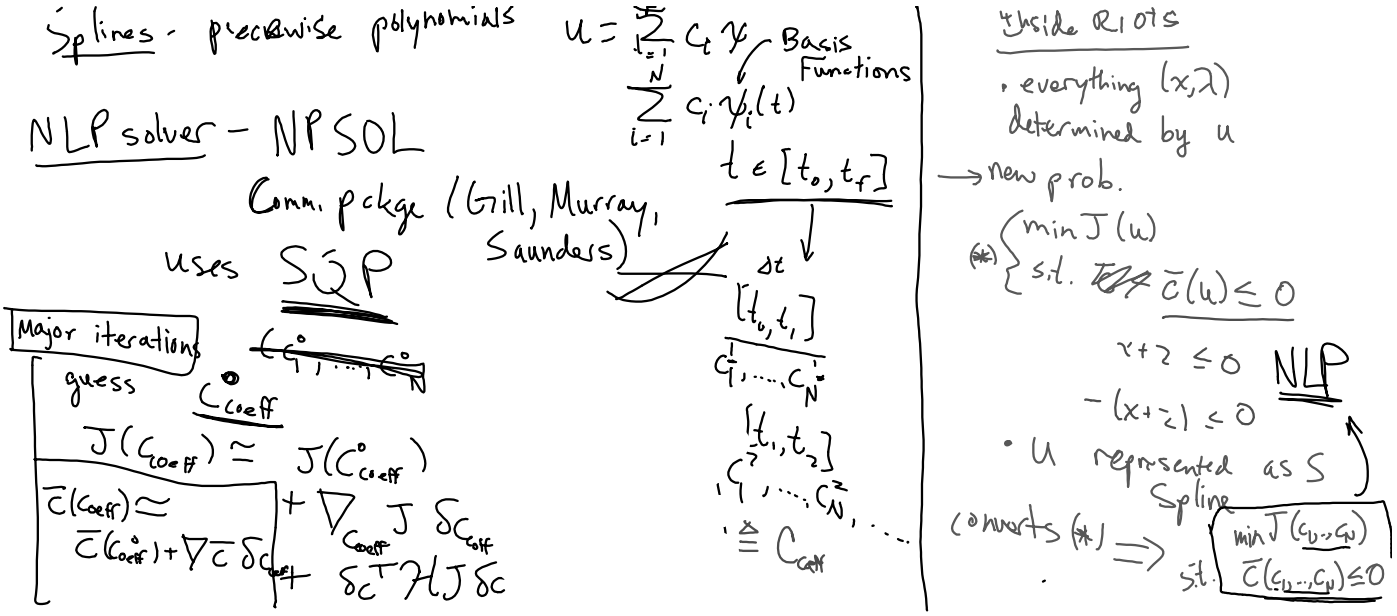
$C(x,u,t) \le 0$, $t \in [t_0, t_f]$

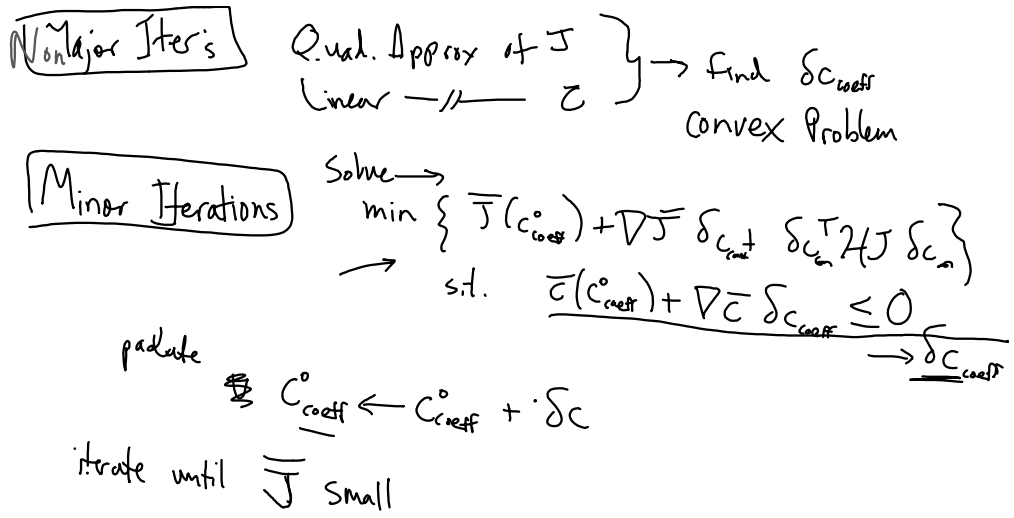
$$\begin{bmatrix} \psi_1(x(t_f)) \le 0 \\ \psi_2(x(t_f)) = 0 \end{bmatrix}$$

Issue with "$\le$" is → introduces combinatorial problem

Sketch:

- gradient algorithm
- computes ("$u(\cdot)$") gradient by the Adjoint ($\delta \bar{J} = \int_{t_0}^{t_f} H_u \delta u \, dt$)
- "u" parmet. as Splines
- RK ode solver


Splines - piecewise polynomials
$u = \sum_{i=1}^{N} c_i \psi_i$ Basis Functions
$\sum_{i=1}^{N} c_i \psi_i(t)$
$t \in [t_0, t_f]$
NLP solver - NPSOL
Comm. pckge (Gill, Murray, Saunders)
uses SQP
Major iterations
guess C^0_{Coeff}
$J(C_{Coeff}) \simeq J(C^0_{Coeff}) + \nabla_{Coeff} J \, \delta C_{Coeff}$
$\bar{c}(C_{Coeff}) \simeq \bar{c}(C^0_{Coeff}) + \nabla \bar{c} \, \delta C_{Coeff}$
$+ \delta C^T \mathcal{H} J \, \delta C$
st
$[t_0, t_1]$
$C_1^1, \ldots, C_N^1$
$[t_1, t_2]$
$C_1^2, \ldots, C_N^2, \ldots$
$\triangleq C_{Coeff}$
Inside Riots
• everything (x, λ) determined by u
→ new prob.
(*) $\min J(u)$ s.t. $\bar{c}(u) \le 0$
$x + 2 \le 0$
$-(x+2) \le 0$
NLP
• u represented as Spline
converts (*) ⇒
$\min J(C_1, \ldots, C_N)$ s.t. $\bar{c}(C_1, \ldots, C_N) \le 0$


Major Iter's
Quad. Approx of J
Linear —//— c
→ find δc_coeff
convex Problem
Minor Iterations
Solve →
min { J̄(c°_coeff) + ∇J̄ δc_coeff + δc_coeff^T ∇²J δc_coeff }
s.t. c̄(c°_coeff) + ∇c̄ δc_coeff ≤ 0
→ δc_coeff
update
c°_coeff ← c°_coeff + δc
iterate until J̄ small

Nonlinear Trajectory Generation

uses collocation: Represent ~~(x, u)~~ z [Those states & inputs necessary to recover all (x,u) algebraically]

is B-splines: piecewise polynomials defined over intervals (time) by knot points.

idea → look for soln in subspace of z (coefficients)

$$\dot{} = \begin{cases} \dot{x} = \cos\theta\, u_1 \\ \dot{y} = \sin\theta\, u \\ \dot{\theta} = \tan\phi\, u_1 \\ \dot{\phi} = u_2 \end{cases} \qquad \begin{matrix} x \\ y \\ \theta \\ \phi \\ u_1 \\ u_2 \end{matrix} = \sum c_i \psi_i(t) \Big]_z \qquad \frac{\dot{x}}{\dot{y}} = \frac{\cos\theta}{\sin\theta} \rightarrow \theta, u_1$$