Is a Good NLP all you Need to Solve Optimal Control Problems?

John T. Betts

BOEING

Good Software/Algorithms to Solve NLP

Find *Variables* $\mathbf{x}^\mathsf{T} = (x_1, \ldots, x_n)$
to minimize the *Objective*

$$F(\mathbf{x})$$

subject to *Constraints*

$$\mathbf{c}_L \le \mathbf{c}(\mathbf{x}) \le \mathbf{c}_U.$$

Want to Solve Optimal Control Problem

Find *Control Functions* $\mathbf{u}(t)$ to minimize

$$J = \int_{t_I}^{t_F} w\left[\mathbf{y}(t), \mathbf{u}(t), t\right] dt.$$

subject to constraints over the domain $t_I \leq t \leq t_F$

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), t]$$

$$\mathbf{0} \leq \mathbf{g}[\mathbf{y}(t), \mathbf{u}(t), t]$$

and boundary conditions

So What's the Rub?

The NLP Works with a **Finite** Set of Variables $\mathbf{x}$ and Functions $F(\mathbf{x})$, $\mathbf{c}(\mathbf{x})$

$\vdots$

But Optimal Control is an **Infinite** Dimensional Problem; i.e. the functions $\mathbf{u}(t)$ and $\mathbf{y}(t)$

How do we formulate the problem?

Shooting Methods

"Eliminate" Infinite Dimensional Problem by solving

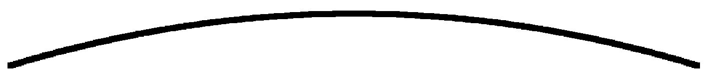
$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), t] \qquad \text{and/or} \qquad \begin{aligned} \dot{\mathbf{y}} &= \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), t] \\ \mathbf{0} &= \mathbf{g}[\mathbf{y}(t), \mathbf{u}(t), t] \end{aligned}$$

The NLP involves the **Finite** Set of Boundary Values

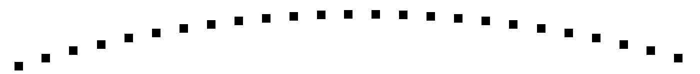
BVP can be very nonlinear
ODE or DAE can be very unstable
ODE error control at suboptimal points—inefficient
Path inequalities cumbersome (impractical?)
Shooting for Control $\Longleftrightarrow$ GRG for NLP

Discretization Methods

Variables



$$[\mathbf{y}(t), \mathbf{u}(t)]$$



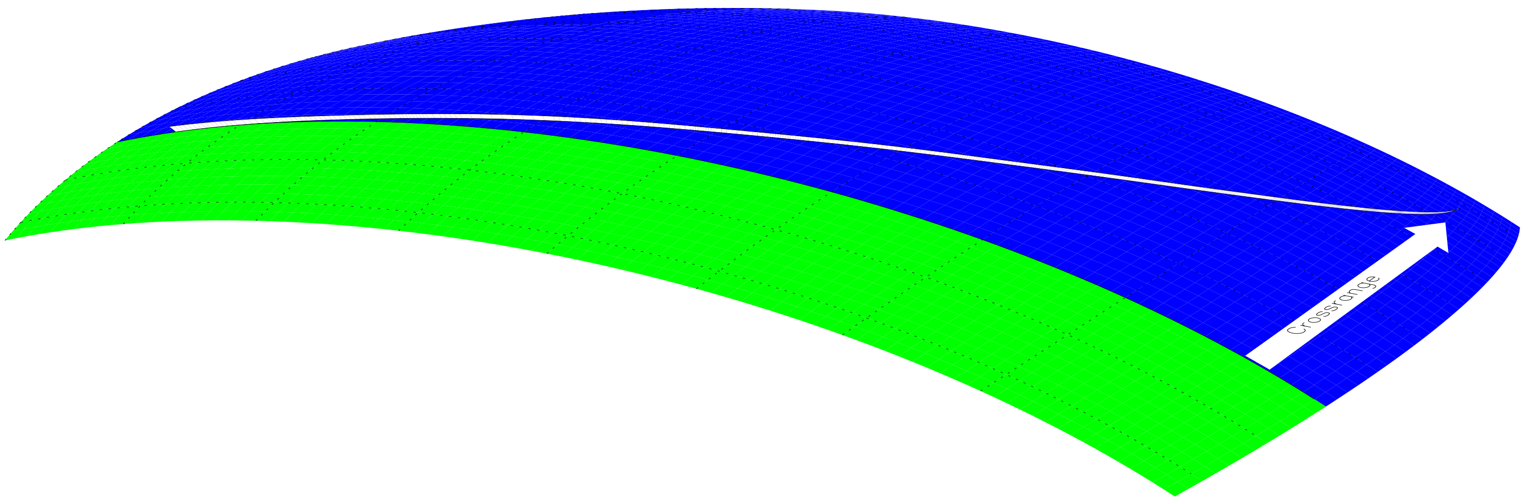
$$\mathbf{x} = [\mathbf{y}_1, \mathbf{u}_1, \ldots, \mathbf{y}_M, \mathbf{u}_M]^\top .$$

Constraints

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), t]$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{h_k}{2}\left(\mathbf{f}_k + \mathbf{f}_{k+1}\right)$$

Maximum Crossrange Reentry



Choose steering $\mathbf{u}(t)$ to maximize crossrange and satisfy

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), t] \qquad \text{Eqn. Motion}$$

$$\mathbf{0} \leq \mathbf{g}[\mathbf{y}(t), \mathbf{u}(t), t] \qquad \text{Heat Limit}$$

An Experiment

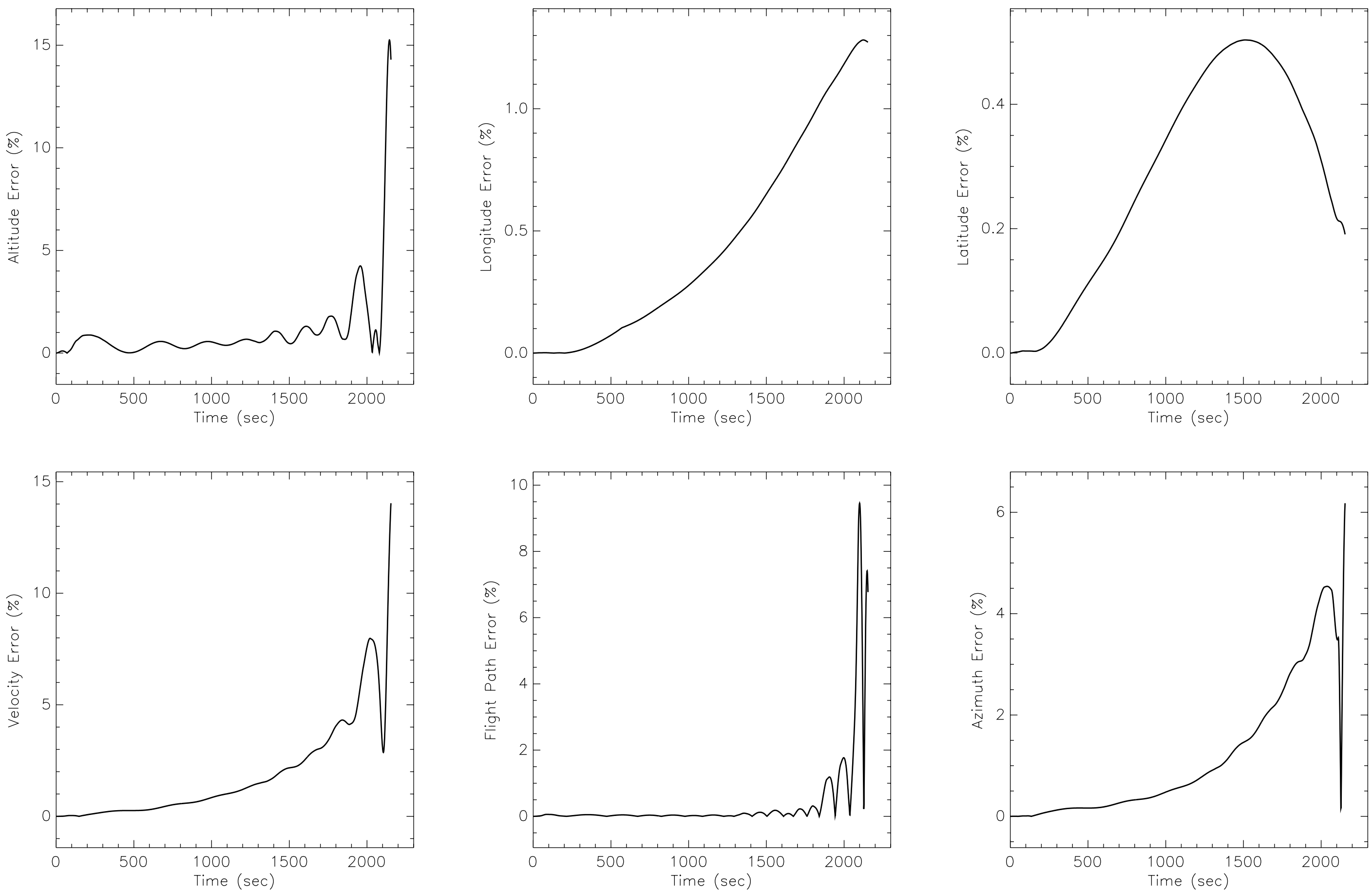
- Use your favorite NLP to solve the discrete problem using
  - Trapezoidal discretization with
  - 50 equally spaced grid points.

- Numerically integrate the ODE's using the "solution" $\widehat{\mathbf{u}}(t)$, i.e.

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \widehat{\mathbf{u}}(t), t]$$

What happens?

Trajectory Error

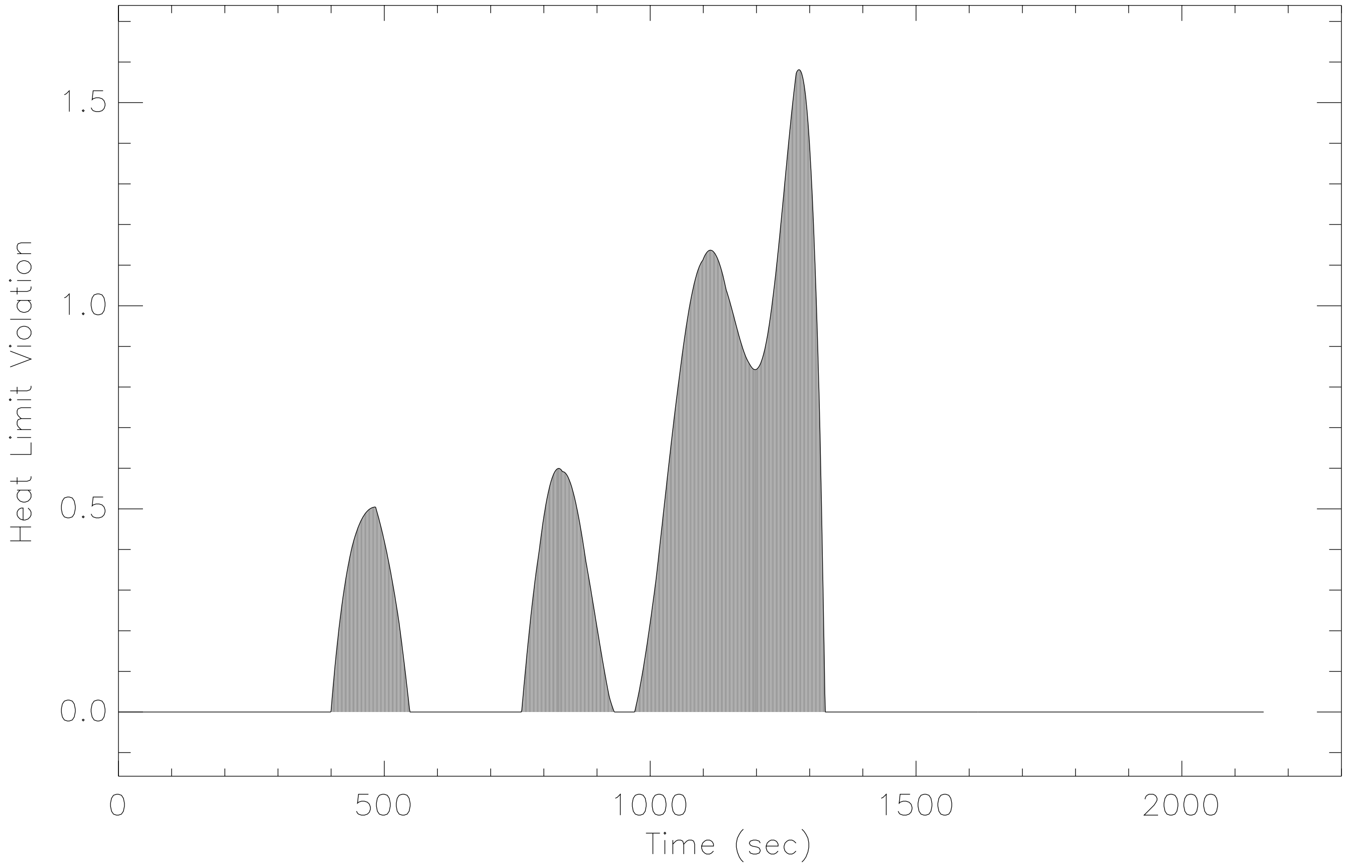


Altitude .... (93343 ft, not 80000)

Velocity .... (2908 fps, not 2500)

Flight Path Angle .... (-9.52 deg, not -5)

Heat Constraint Error



Bad News!

Observation 1

Question: Is a good NLP all you need to solve optimal control problems?

Answer: NO! Discretization Error Must Be Addressed.

An Optimal Control Algorithm

Direct Transcription Transcribe the optimal control problem into a nonlinear programming (NLP) problem by discretization;

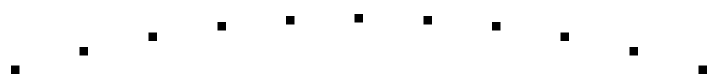
Sparse Nonlinear Program Solve the sparse NLP

Mesh Refinement Assess the accuracy of the approximation (i.e. the finite dimensional problem), and if necessary refine the discretization, and then repeat the optimization steps.

SNLP: Sequential Nonlinear Programming

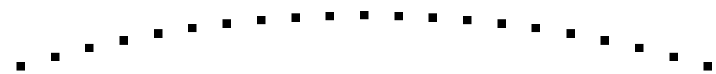
Sequential Nonlinear Programming

Coarse Grid



$$\mathbf{x}_c = [\mathbf{y}_1, \mathbf{u}_1, \ldots, \mathbf{y}_m, \mathbf{u}_m]^\top$$

Fine Grid



$$\mathbf{x}_f = [\mathbf{y}_1, \mathbf{u}_1, \ldots, \mathbf{y}_M, \mathbf{u}_M]^\top$$

NLP problem size grows—typically $M > m$

Question: How do we efficiently solve a **sequence** of NLP's?

Answer: Use coarse grid information to "Hot Start" fine grid NLP

Estimating Variables for SNLP

SQP Algorithm

high order interpolation of coarse grid solution
consistent with discretization formula
(e.g. collocation polynomial)

very good guess

Interior Point Algorithm

must be *feasible* $\Longleftrightarrow$ barrier algorithm perturbs guess
not consistent with coarse grid discretization formula

may be a poor guess

Estimating Multipliers for SNLP

SQP Algorithm

use "cold start" (first order) estimates based on variables

Option: interpolate coarse grid (adjoints) but
NLP Multipliers may not converge to adjoints and/or
discretized path constraints may violate LICQ
(linear independent constraint qualification)

good active set guess $\Longleftrightarrow$ more efficient

Interior Point Algorithm

must be dual *feasible* $\Longleftrightarrow$ barrier algorithm perturbs guess

must be consistent with initial barrier parameter

"active set" not an issue

may be a poor guess

Linear Algebra for SNLP

SQP Algorithm

active sets and iterative methods not practical

direct factorization limits largest problem size

Interior Point Algorithm

iterative methods possible

coarse to fine grid preconditioners

potential for very large problem size

Performance Comparison

SQP Algorithm

k	M	NGC	NHC	NFE	ϵ	Time (sec)
1	50	27	18	901	8×10^{-1}	3.4
2	66	14	12	566	8×10^{-2}	2.4
3	66	10	8	386	3×10^{-2}	4.7
4	95	19	17	791	6×10^{-3}	13.0
5	100	22	20	926	1×10^{-3}	15.2
6	105	9	6	313	1×10^{-4}	8.0
7	110	4	2	116	2×10^{-5}	6.0
8	219	10	8	386	4×10^{-6}	35.5
9	228	3	1	71	7×10^{-7}	19.5
10	455	5	3	161	4×10^{-8}	81.7
Total	455	123	95	4617		189.

Barrier Algorithm

k	M	NGC	NHC	NFE	ϵ	Time (sec)
1	50	74	69	3289	7×10^{-2}	8.3
2	86	10	3	246	5×10^{-3}	1.5
3	86	29	27	4679	9×10^{-5}	19.4
4	171	9	5	978	3×10^{-6}	9.3
5	323	44	40	7001	1×10^{-7}	125.
6	573	19	16	2824	6×10^{-8}	102.
Total	573	185	160	19017		266.

Analysis of Performance

Expected Behavior

As Mesh Size Increases . . . .

- SQP becomes more efficient because it exploits a good guess
- Barrier method does not exploit good guess

Unexpected Behavior

Why do SQP and Barrier methods have
Different Number
of Refinement Iterations?

SQP vs Barrier Coarse Grid Solution

Final Condition	SQP	Barrier
Time (sec)	2153.85	2186.11
Altitude (ft)	80000.0	80000.0
Longitude (deg)	50.9380	51.6154
Latitude (deg)	30.3827	30.5691
Velocity (fps)	2500.00	2500.00
Flight path angle (deg)	-5.00000	-5.00000
Azimuth (deg)	11.5361	7.64752
Angle of Attack (deg)	5.44440	17.4327
Bank Angle (deg)	-90.0000	-2.04101

Same NLP Problem (Trapezoidal, 50 grid points)
Different (Wrong) Answers!

The State of Affairs

Must Address Interaction Between Optimization and Discretization

Nonunique, Local Solutions for the Same Discrete Subproblem

SQP Algorithm Can Exploit Good Guess

Barrier Algorithm Can Exploit Iterative Linear Algebra