High-Performance Computational Optimal Control via Eliminating Control Variables

Dan Moerder, LaRC

8 October, 2008

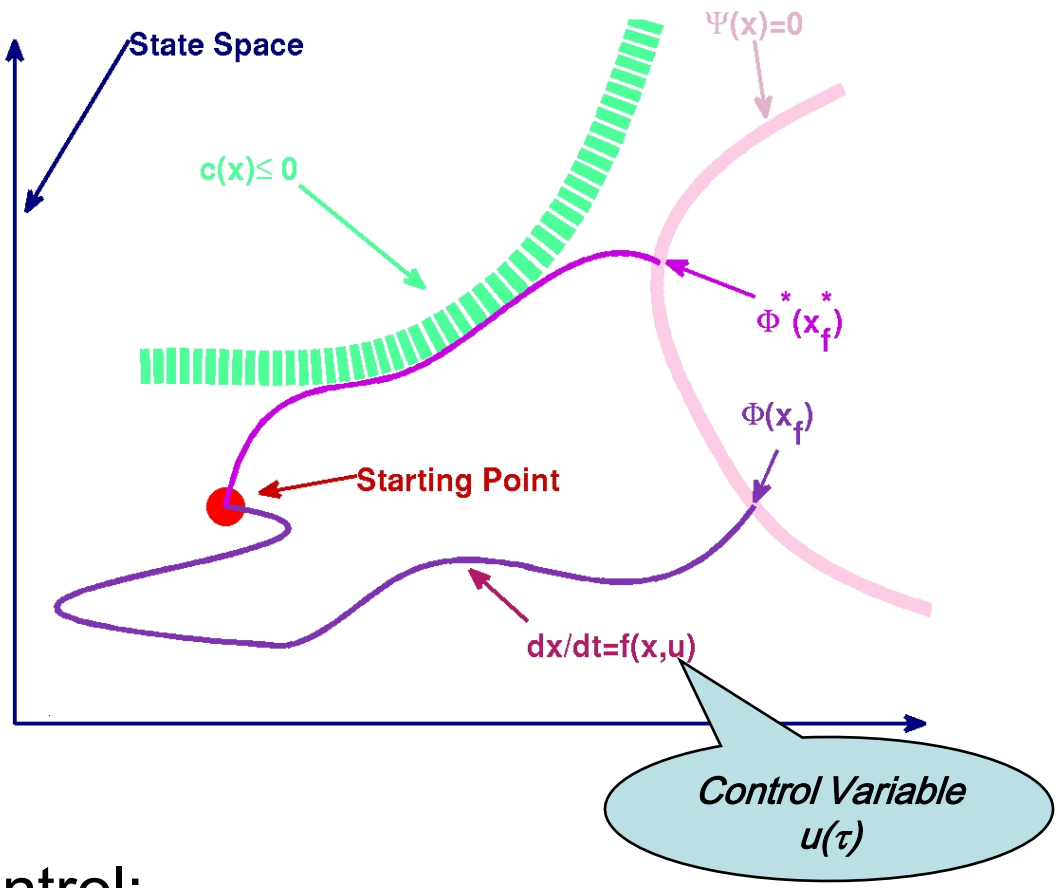
Purpose of Talk

- Motivate uses for Computational Optimal Control (COC) in hypersonic flight systems
- Discuss challenges posed by the presence of control variables in COC problems
- Describe an approach for mitigating these challenges

Structure of Talk

- What is COC?
- What, in principle, would we like to use it for?
- Why can't we?
  - *Control variables aren't the whole problem, but they certainly don't help…*
- Discrete "maximum principle" for control elimination
- An approach for forcing convex state rates
- Status of Project, and a summary of the importance of this work to hypersonics.

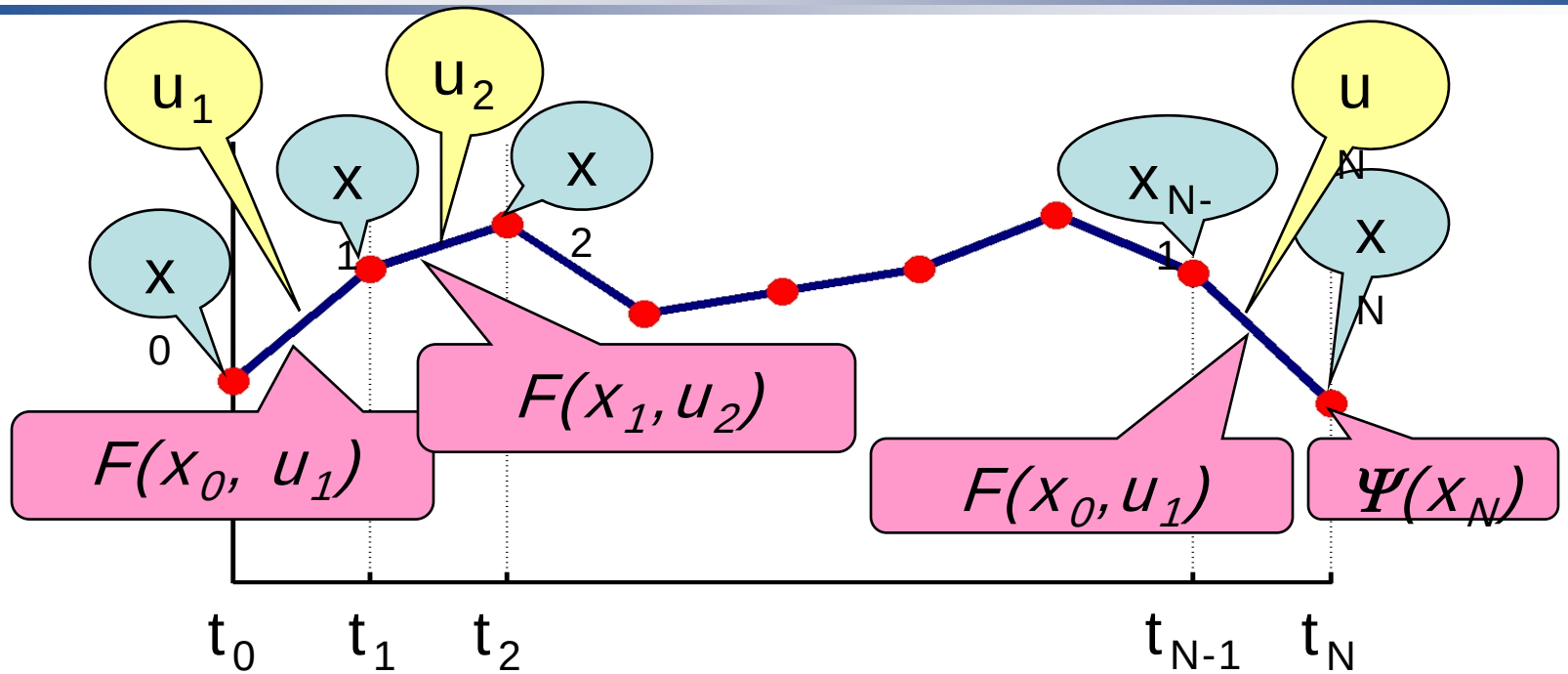
What is Optimal Control?



Optimal control:

$u^*(\tau)$, $t \le \tau \le t_{final}$ *such that* $\Phi^* \le \Phi$ *for all possible* $u(\tau)$

What is Computational Optimal Control?

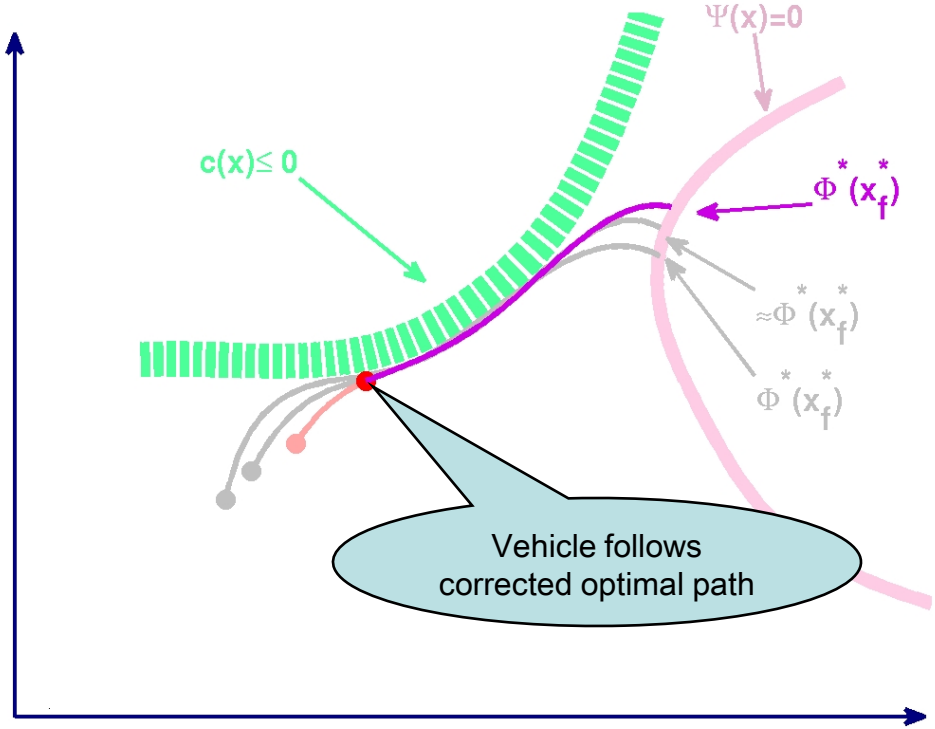


- Set up discrete-time representation of system.
- Choose $x_0, x_1, \ldots, x_N$ and $u_1, u_2, \ldots, u_N$ to minimize Φ while satisfying the discretization expressions (the Fs) and boundary conditions (Ψ)
  - *Use a generic Nonlinear Programming (NLP) code for this*
- As N increases, the "u" variables increase the problem's degrees of freedom

For What, in Principle, Can COC Be Used?

Hypersonic Trajectory Control and Control-Related Analyses

- Optimality is important in hypersonics because there is
  - *little excess performance, i.e., the vehicle is difficult to close.*
  - *complexity ⇒ Engineering intuition is only a weak guide*
- Application: Feedback Control:



For What, in Principle, Can COC Be Used?

Hypersonic Trajectory Control and Control-Related Analyses

- Optimality is important in hypersonics because there is
  - *little excess performance, i.e., the vehicle is difficult to close.*
  - *complexity ⇒ Engineering intuition is only a weak guide*
- Another Application: System Design
  - *The MDAO community uses COC by including system configuration parameters in the COC problem statement.*
  - *Example:*
    - Minimize initial fuel load to achieve desired fuel reserve at target condition.
    - Size of initial fuel load *ideally* determines tankage, OML, inert mass, etc.
  - *This is not trivial to do with any modelling fidelity in the current state of the art.*
  - *Although this project is focussed on COC for hypersonic trajectory control, successful concepts and algorithms here may be useful for the Hypersonics MDAO effort.*

Terrifyingly Naïve Assumptions (TNAs)

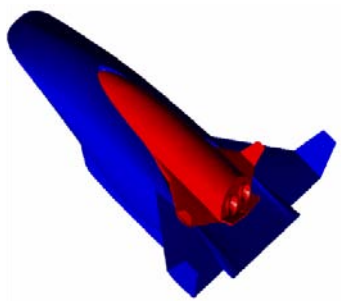
- The example applications from the previous slide assume
  - *The computations leading to the COC solution are reliable and timely.*
  - *The COC formulation and solution algorithm can cope with realistically complicated models.*

Well, What About Our TNAs?

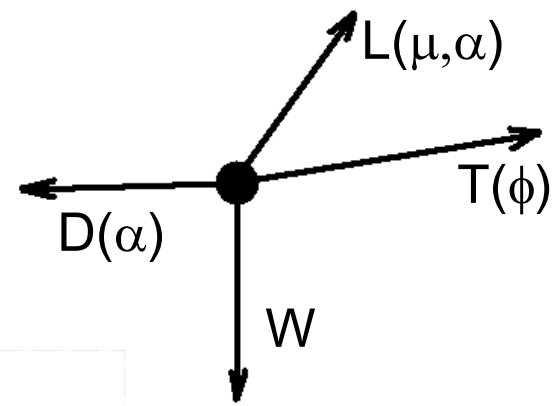
Caveats:

- COC is not "typically" tolerant of
  - *Redundant control effectors*
  - *Slow/fast dynamics, e.g. energy vs rigid body rotations*

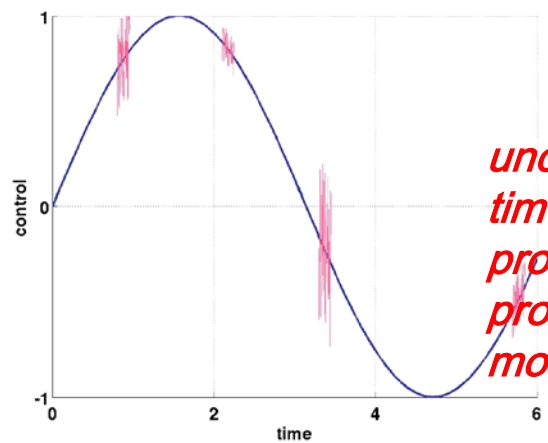
⇒ "Point mass" models are typical



trim analysis



- COC is **definitely** not tolerant of chattering control



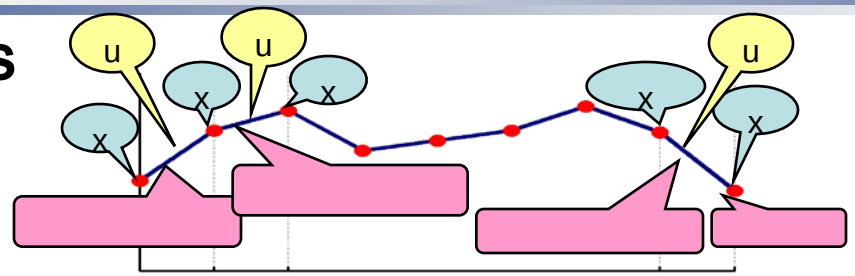
Chattering is not uncommon in continuous-time optimal control problems, and arises from properties of the dynamical model

Implications of TNA Failure

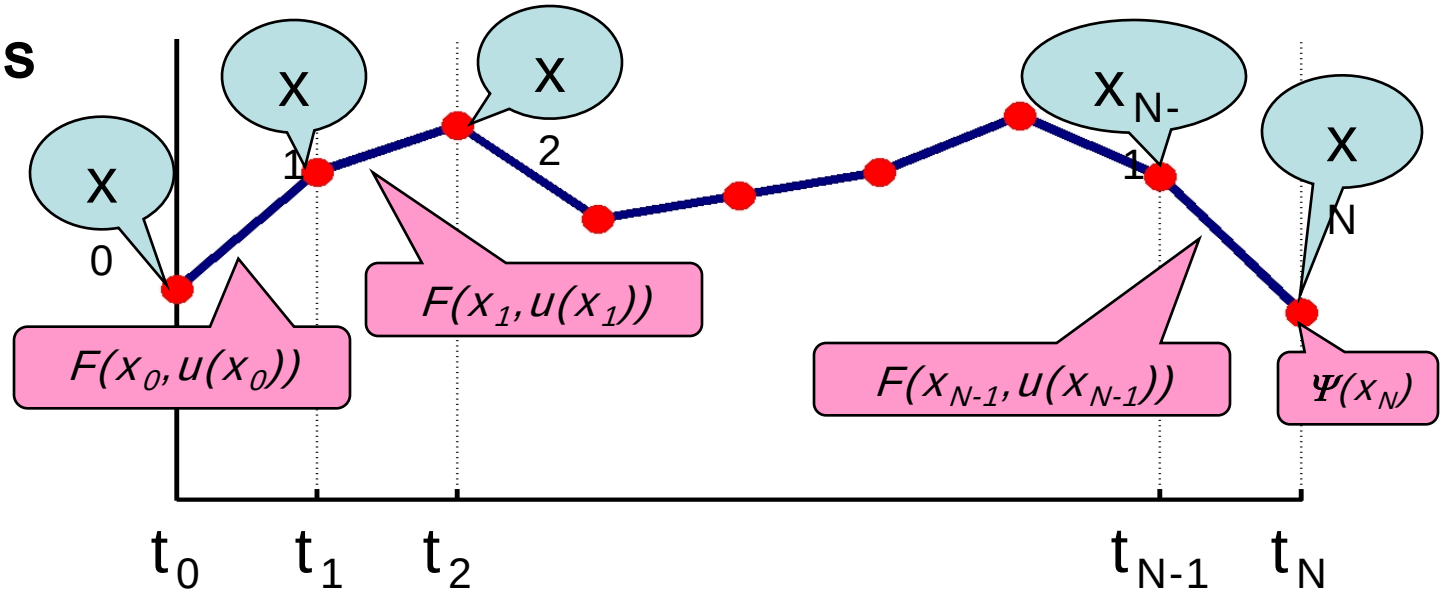
- For Feedback Control,
  - *Use of point mass dynamics may require pre-computation of trim models for "all possible" cases…*
    - At the very least, covering all of the possible aero-thermo-elastic-propulsive combinations will be burdensome.
    - This is very vulnerable to Murphy's Law.
  - *In order to ensure convergence of control computations, all models would have to be examined and scrubbed to ensure that chattering won't occur.*
- For System Design,
  - *Lack of clear, simple, link between "full" dynamics and point-mass model requires trim-oriented massaging of the model throughout the flight envelope at each iteration of the optimization process.*
- *Not good. What can we do?*

Computational Optimal Control Without Controls

Instead of this



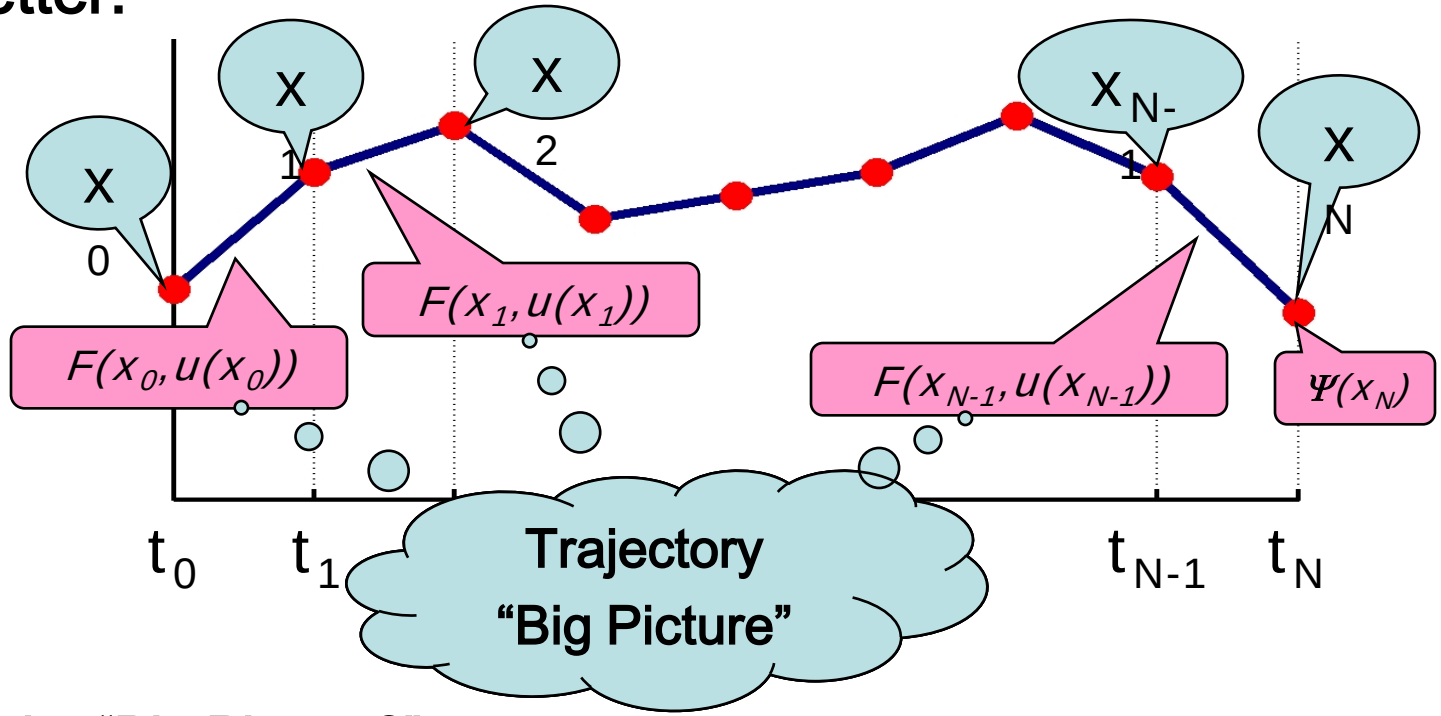
Try this



- Assume that each control is a known function of the state
  - *Controls can be computed via separate optimization for each given value of state – output of process is state rate in "F".*
  - *Trajectory optimization problem now only solves for states.*
- This won't work as drawn…

COC Without Controls, continued

This is better:



- What is the "Big Picture?"
  - *The u(x) computation needs to know what "u" is going to be used for…*
  - *In other words, what is the best state rate*

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

for the "top-level" x-only trajectory optimization?

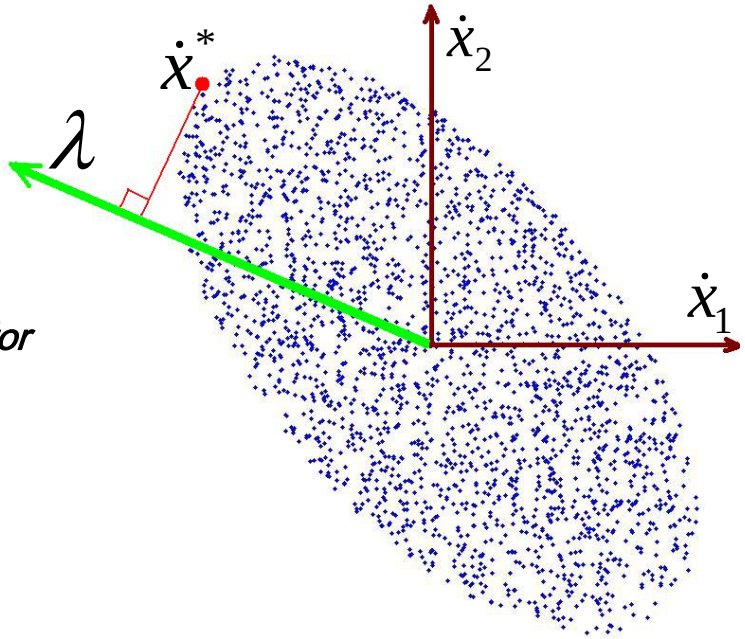
Pontryagin's Maximum Principle Provides the Clue

- In continuous-time optimal control, choose u* to maximize

$$H = \lambda^T f(x,u)$$

where $\lambda(t)$ is a sensitivity function associated with the system dynamics, $\Phi(x_f)$, and $\Psi(x_f)$

- If you're using an augmented Lagrangian code to solve the NLP problem (*of course you are*,) you have $\lambda(t)$ as part of the optimization machinery.
- Here's what we have:
  - *Ellipse is set of all achievable state rates (hodograph) for the state at a given point in the trajectory*
  - *Optimal xdot has largest projection onto λ–vector*
  - *Optimal control value is any value of u that produces the optimal xdot.*



Several Observations

- This sounds very simple, but we can make it complicated (in a good way).
- The vector λ merely expresses what the top-level trajectory optimization process wants from u.
  - *The direction of λ indicates where the state rate should mostly point,*
  - *The length of λ indicates how urgently the top-level optimization wants the state rate to point there.*
- The vector λ can be thought of as more of a guideline than a command.
- The control can be computed to achieve any goal we desire, as long as
  - *The state trajectory is able to meet its constraints,*
  - *The resulting u(x) is twice differentiable.*
- The plant model inside the control optimizations can be different from that used in the top-level optimization
  - *It can be more detailed,*
  - *It can be just plain different.*

Exploiting This Notion

- Put rigid-body dynamics into the Control Optimization Model (COM)
  - *trim the vehicle,*
  - *and appropriately adjust the hodograph.*
- Put aero-thermo-elastic-propulsion dynamics into the COM,
  - *compute inner-loop controllers,*
  - *then compute RMS response for assumed disturbances,*
  - *then, compute RMS aero and thrust properties,*
  - *and appropriately adjust the hodograph.*

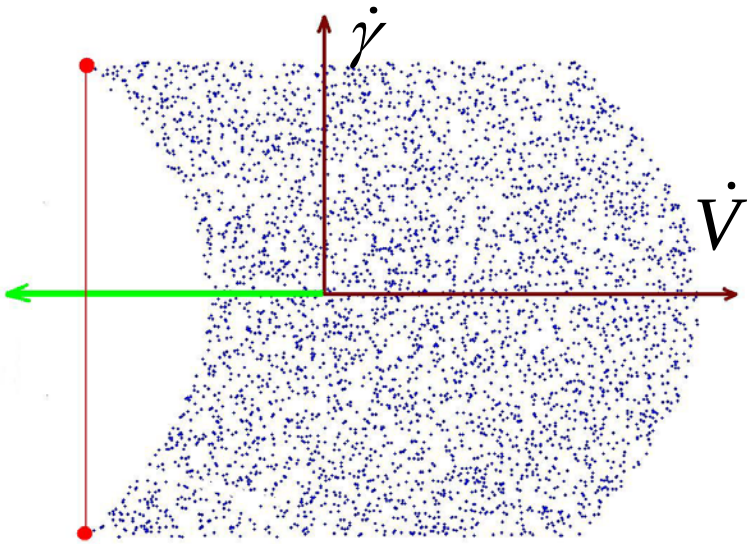
- *This opens the door to using the trajectory optimization for automated vehicle design with realistic dynamical effects included.*
- These are straightforward exploitations of the "inner-loop/outer-loop" architecture described in previous viewgraphs

- *This opens the door to more reliable computations for feedback control*

- Convexify the hodograph!

The Hodograph: Is There a Problem?

- Consider aircraft V,γ dynamics, controlled by T,L:
  - *First λ scenario: convex hodograph region, unique optimal state rate…*
  - *Second λ scenario: concave hodograph region, two separate optimal state rates*
  - *Third λ scenario: singular hodograph region, many equivalent state rates*



- The second, chattering control scenario is not uncommon for aerospace problems
  - *They can occur because of funny OML contours (ALS program asymmetric booster had concave hodograph)*
  - *They can occur because of glitches in tabular data*
  - *They can occur because of parabolic drag polar (TAEM maneuvers)*
- Third scenario is bad, too; but not as much of a problem as chattering.

Point-Wise Convexification Algorithm

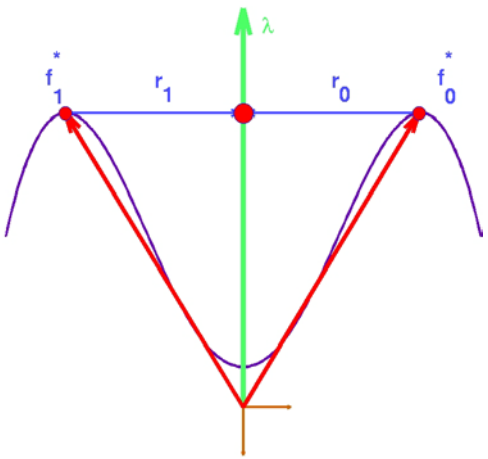
- Find an optimal rate f*
- Look for another, subject to the constraint that

$$\langle r_0, r_1 \rangle \leq 0$$

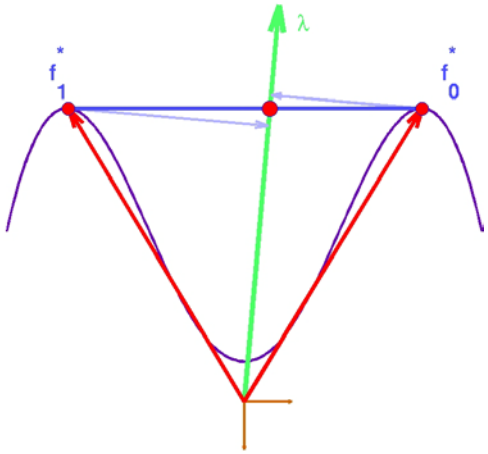
- If there's a second f* of equal optimality such that

$$\langle r_0, r_1 \rangle < 0$$

  then average the two rates



- What if there's only one optimum, but there's another, different, rate that gives similar performance?
- Take the convex combination of the two rates
- *This is only possible because*
  - *We have the λ vector*
  - *We are hiding our suboptimal shenanigans from the top-level optimization.*



Project Status

- The MADS trajectory optimization code has been completely rewritten to support
  - *Hierarchical optimization, e.g. local control optimization,*
  - *High-fidelity differentiation,*
  - *High-order (RK45) discretization, as an option.*
- This came to roughly 3000 non-comment lines of FORTRAN95 code, with an additional 5000 lines of Matlab and FORTRAN95 test code.
- Point-wise hodograph convexification has been formulated in some detail and Matlab prototyping is underway.
- The details of a scheme for singularity compensation are being worked out.

Impact of This Work on Hypersonics

- Computational Optimal Control (COC) is important to hypersonics because hypersonic vehicles are likely to be complex and performance-limited for the foreseeable future.
- This research has produced an apparently novel approach for enhancing the reliability of COC numerics.
- By converting the control variables to functions of the state at the interface between the plant dynamics and the state trajectory, we
  - *permit arbitrarily large and complicated computations inside the plant dynamics "blocks" at each instant along the COC trajectory.*
  - *permit distorting the plant dynamics in an orderly way to make the optimal control numerics benign without distorting the relevant performance capability of the plant.*
- This work enables optimal feedback guidance with onboard identification of detailed plant parameters, or reconfiguration after unexpected changes in the system.
- This work, when sufficiently mature, may provide an attractive leveraging opportunity for use by MDAO for automating optimal system design.