Bipeds: Simulation & Optimization

Discontinuities Make Trouble (examples)

Matthew Kelly & Andy Ruina   Cornell University

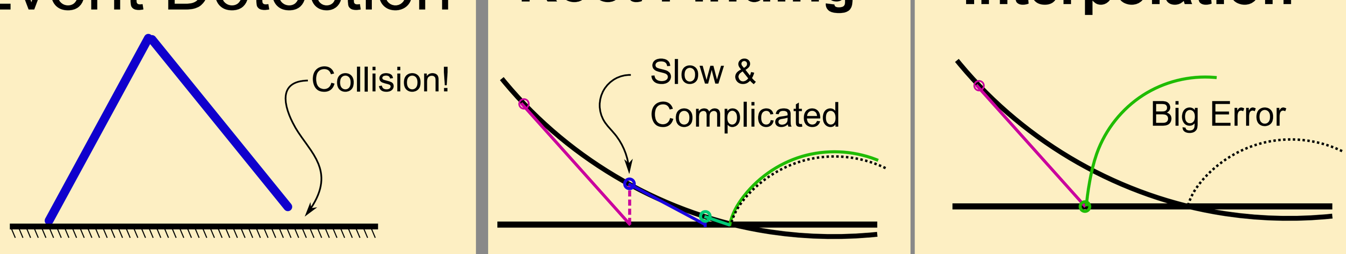
0. What we want
(... for biped controller design)

1) Optimal trajectories
2) ≈ Optimal policy
3) Test and improve controller

Challenges = discontinuities
- Heel-strike
- Toe-off
- Coulomb friction
- If statements
- State machines
- *etc.*

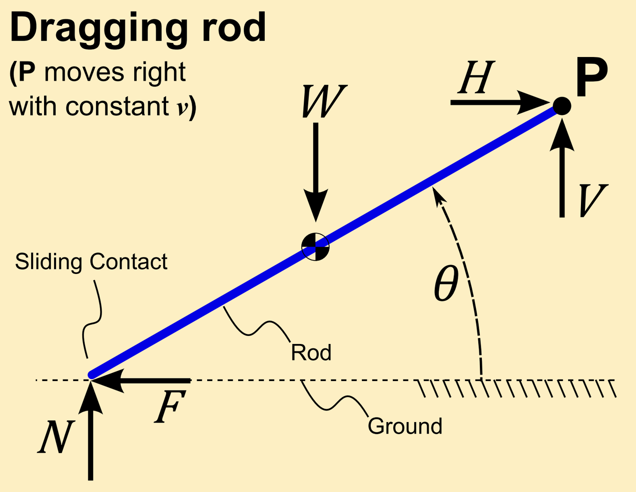
Tools available
- Trajectory optimization (GPOPS or home made)
- Simulation
- Optimizers
  - → Gradient-based (SNOPT, IPOPT, FMINCON)
  - → Heuristic (CMAES, Genetic Algorithm, Random Search)

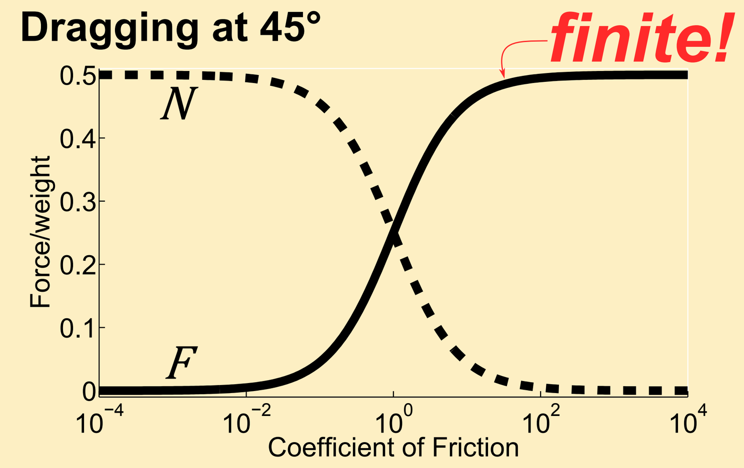
1. Simulation: Precise vs. Fast



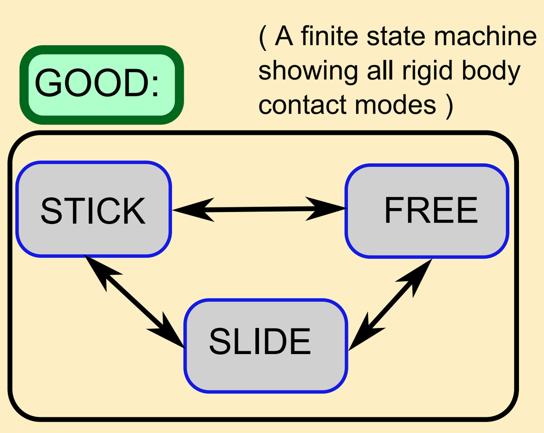
	Complex Slow	Simple Imprecise
Event Detection	**Root Finding**	**Interpolation**
Integration	High-Order $\varepsilon \sim (\Delta t)^n$	Low-Order $\varepsilon \sim \Delta t$
Dynamics (and Control)	Smooth -- or add events --	Arbitrary eg. *abs(x)* is ok

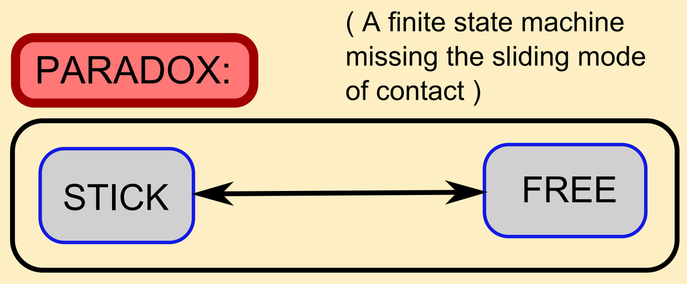
2. Infinite Friction ⟷ No-Slip





"No-Slip" is nonsense





⇒ ***Penetration*** (sometimes)
-- or --
negative contact forces.

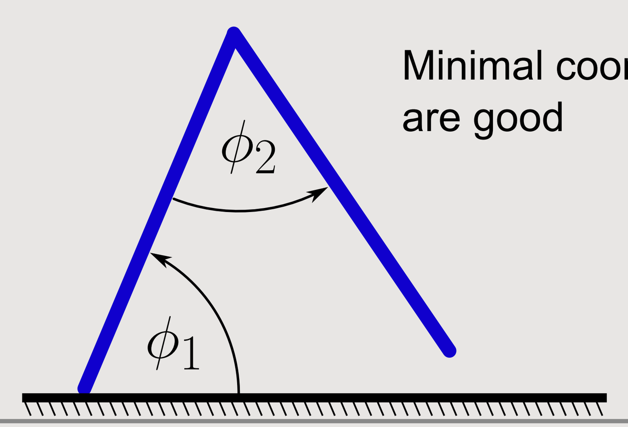
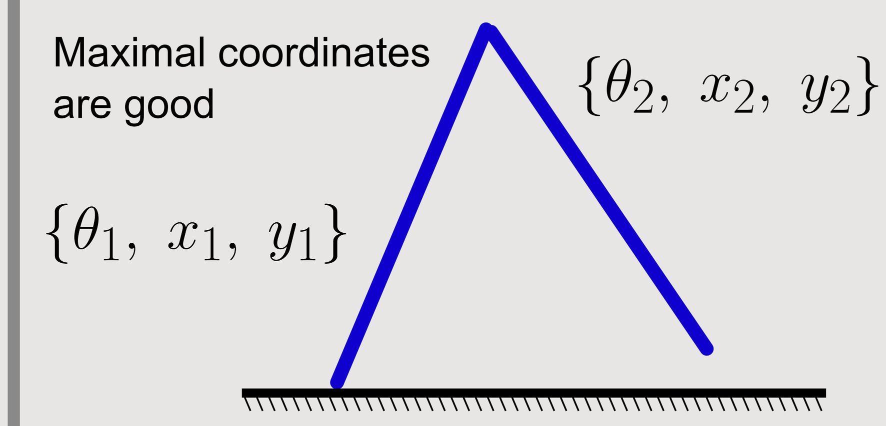
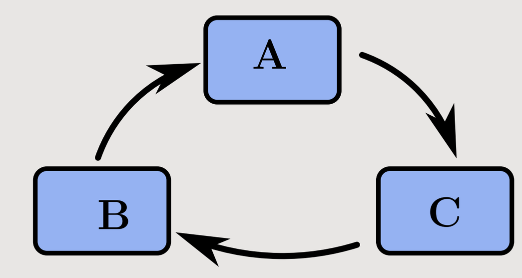
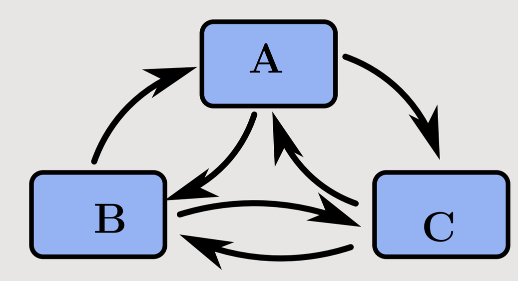
3. No best optimizer

Gradient methods (SNOPT, FMINCON, IPOPT)
- − Need smooth objective
- + Good with constraints (if smooth)
- + Fast convergence (especially if convex and have Jacobian & Hessian)

Heuristic algorithms (CMAES, Genetic Algorithms)
- + Good for slightly irregular objective
- − Not good with constraints
- − Slow convergence (without careful parameter tuning)

Brute force search
- + Always works (eventually)
- − Not good with constraints
- − *Terrible* convergence (unless lucky)

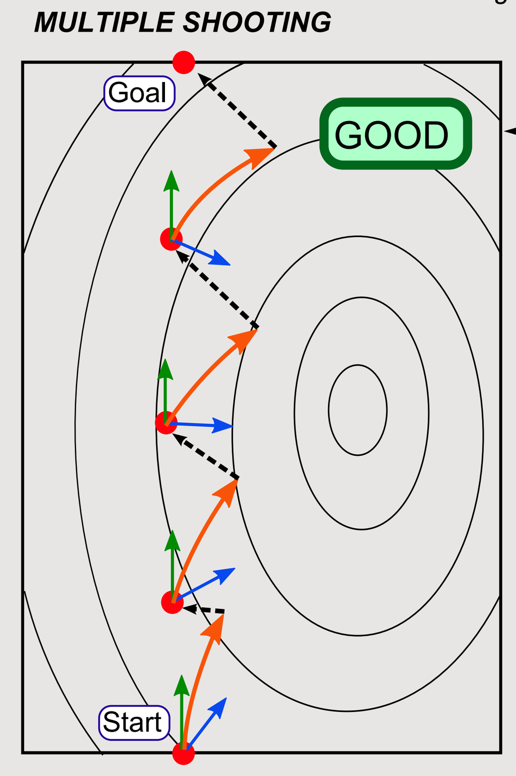
4. Trajectory Optimization ≠ Simulation

Trajectory Optimization	Simulation
Minimal coordinates are good	Maximal coordinates are good
	
 Prescribed event sequence	 Arbitrary event sequence
Control *Solution*: $\vec{u} = \vec{u}(t)$ (scalar)	Control *Policy*: $\vec{u} = \vec{u}(\vec{x})$ (vector)
Smooth and Consistent Objective and Constraints Functions	Arbitrary Dynamics and Control Functions

Our simulation uses a general impulse-based contact solver and has an arbitrary controller.
Our optimization assumes a known sequence of contacts, such as in normal walking.

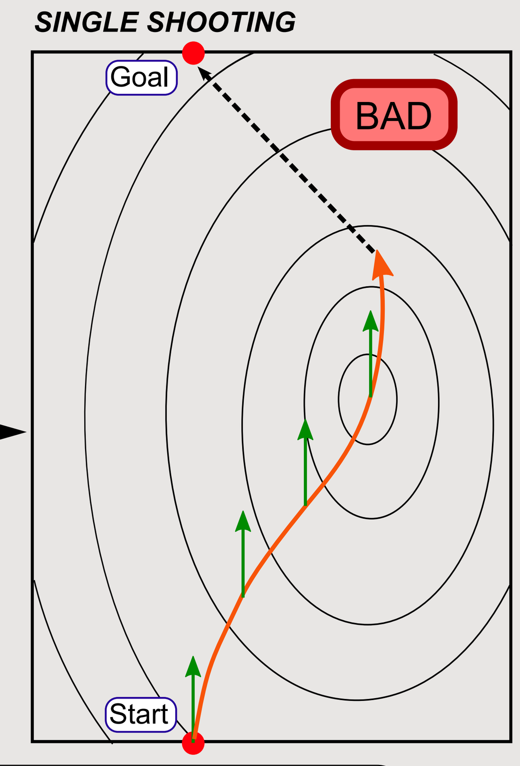
5. Trajectory optimization tricks

"Solving a problem in optimal control ... is not easy" -John Betts 2010



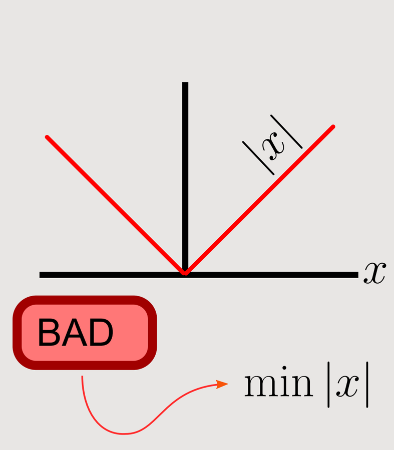
+ MULTIPLE SHOOTING
Many small defects
- Decoupled
- Nearly linear
- Fast convergence

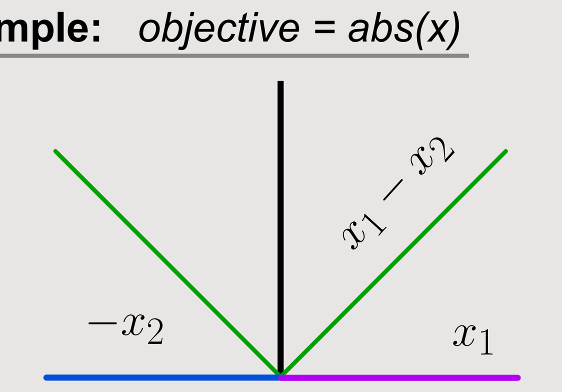
− SINGLE SHOOTING
One big defect
- Fully coupled
- Way nonlinear
- Bad convergence



Defect Vector
Motion over one step
Potential Field Lines
Position in State Space
Actuator Guess
Disturbance (from field)

Discontinuity example: *objective = abs(x)*





$$\begin{aligned} \min:\ & (x_1 + x_2) \\ \text{subject to:}\ & x = x_1 - x_2 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

GOOD

Other tips
- + Use fixed-step integration
  - ⇒ Keeps # gridpoints constant
- + Add regularization to objective
  - ⇒ Forces unique solution
- + Include control in state space
  - ⇒ Keeps all constraints state-based

  ie. state space $\{x,\ u\}$

  $\dot{u}$ = new decision variable

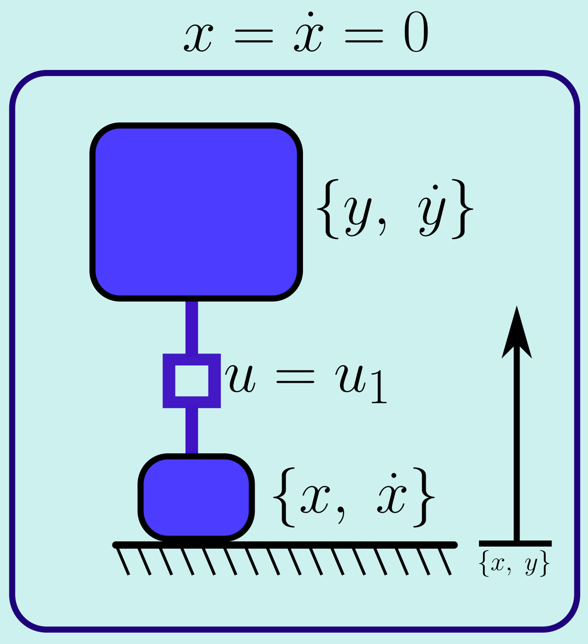
6. Must validate controller liveness

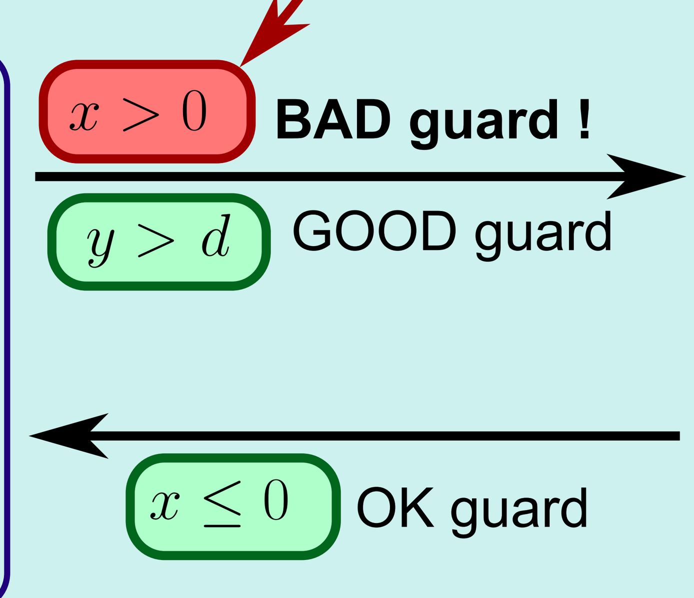
Simple (bad) hopping controller:

```
If (on ground) -- extend leg
Else -- retract leg
```

The 'BAD' guard shown below will never be true. This will prevent the controller from switching, thus it fails the liveness condition (producing a hopping behavior).

On ground
$x = \dot{x} = 0$





In flight
$x > 0$