Hierarchical Biped Control

A Exam

Matthew Kelly

August 4, 2014

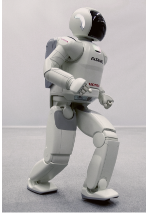
Table of Contents

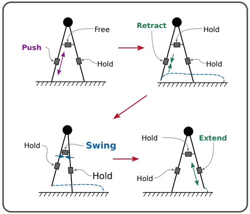
1 Background

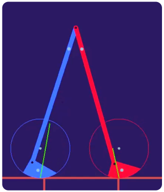
2 My PhD Plan

3 Completed Work

4 Planned Work





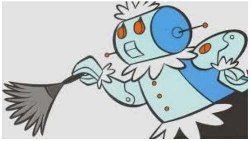


Motivation

Education

Military

Personal Helper



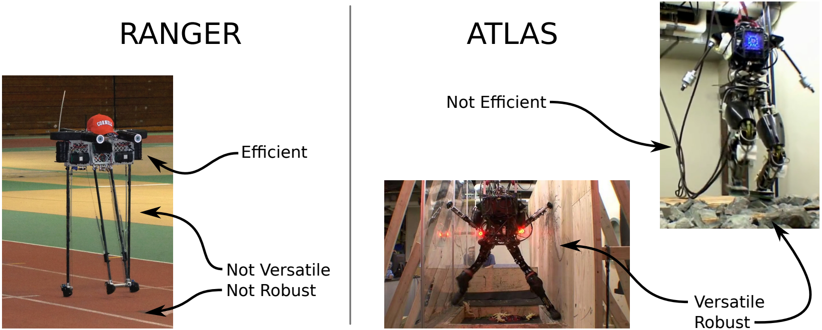
Prosthetics

Disaster Relief

Motivation

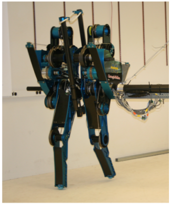
Why are biped robots not widely used?

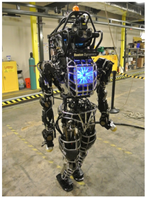
- Modern controllers are not robust, versatile, nor efficient
- Also unsolved: perception, planning, interaction...



Background: Popular Ideas in Locomotion Control

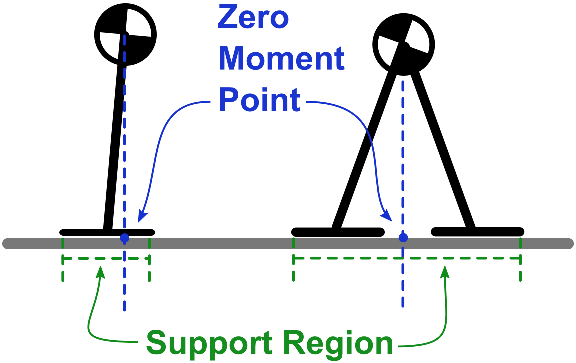
1. Background
   - Zero Moment Point
   - Capture Point
   - Hybrid Zero Dynamics
   - Other Ideas
2. My PhD Plan
3. Completed Work
4. Planned Work



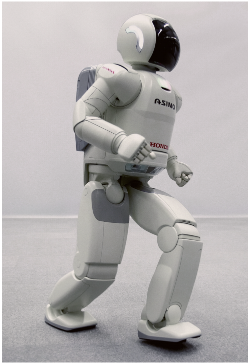


Background - Zero Moment Point (ZMP)

Walking as a perturbation of standing

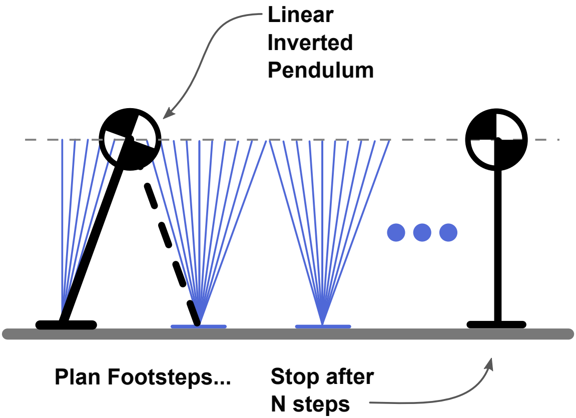


Asimo
→ Honda

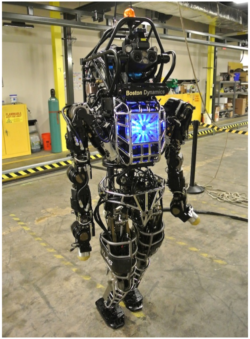


Background - Capture Point

Walking as a perturbation of falling

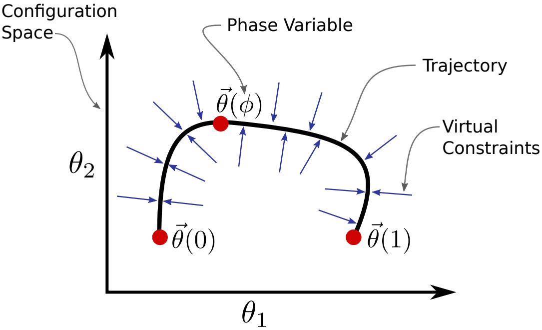


Atlas
→ Boston Dynamics

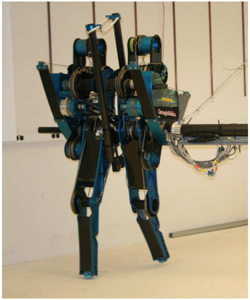


Background - Hybrid Zero Dynamics

Walking as trajectory tracking:

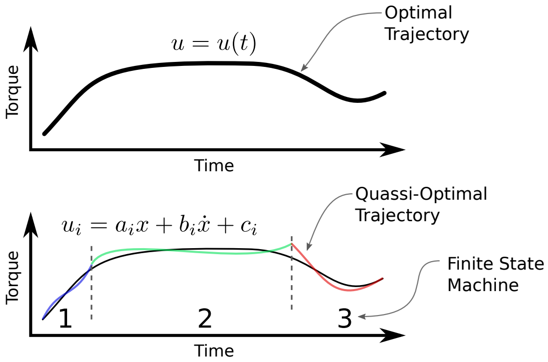


Mabel
$\rightarrow$ U. Michigan



Background - Heirarchial Concurrent FSM

Walking as simple finite state machine:



Ranger

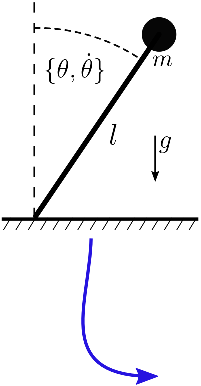
$\rightarrow$ Cornell U.

Other ideas for locomotion controllers

- **Model Predictive Control** - Emo Todorov - U. Washington
- **LQR Trees** - Russ Tedrake - M.I.T.
- **Reflex Control** - Hartmut Geyer - Carnegie Mellon R.I.
- **Neural Nets** - Hod Lipson - Cornell University
- **SIMBICON** - Michiel van de Panne - U. of British Columbia
- **Central Pattern Generators** - Biorobotics Lab - E.T.H.

My Resarch

1. Background
2. My PhD Plan
   - Where do I fit in?
   - Point Mass Walker
   - Heirarchical Control
   - An Outline for my PhD
3. Completed Work
4. Planned Work



Where does my research fit in?

Many ways to make robots walk... but not yet as well as a human

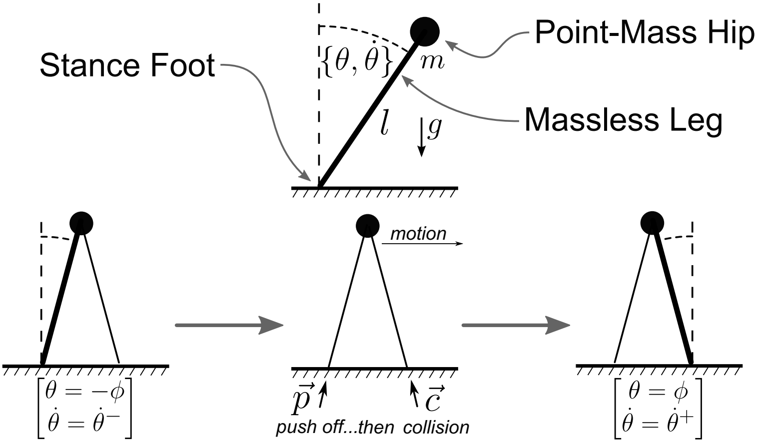
Tendencies in modern robots:

- Constrain away dynamics → reduce robustness and efficiency
- Dependent on ankle torques → reduce robustness
- Focus on efficiency → reduce versatility and robustness
- High bandwidth hydraulics → poor efficiency

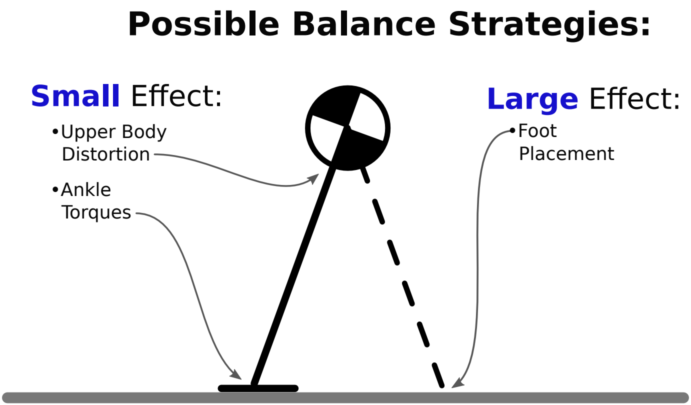
We address these issues by taking advantage of an observation:

- **Bipeds inherently behave like a simple model**

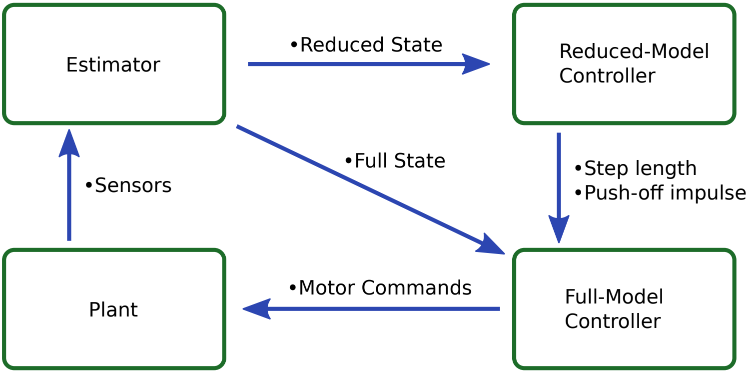
The Point Mass Walker Model



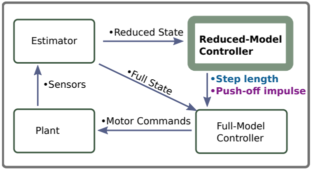
Why Point Mass Model?



Heirarchical Control: Block Diagram



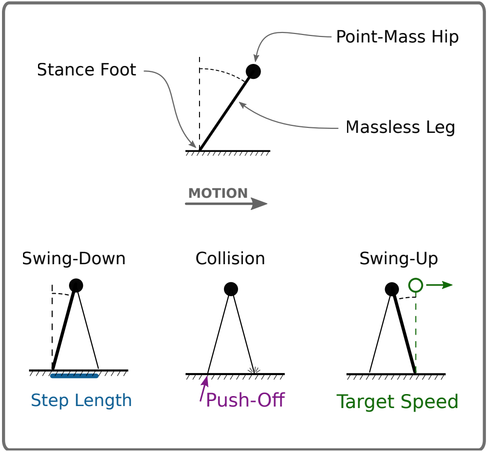
Heirarchical Control: Reduced-Model Controller



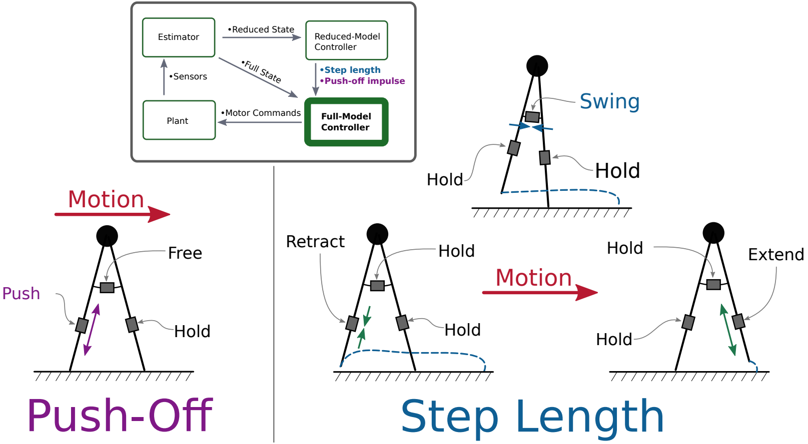
Find:
- step length
- push-off impulse

Such That:
- return to target speed
- minimize energy (?)

Subject To:
- Actuator bounds
- No flying / falling
- Minimum step time



Heirarchical Control: Full-Model Controller



My Contribution:

→ A new control architecture for bipedal locomotion:

Robust:

- Exhaustive testing in reduced-model design
- Full-model inherently behaves like reduced-model

Versatile:

- Reduced-model controller design process is general
- Continuous-time trajectory has simple inputs

Efficient:

- Reduced-model controller design uses energy model
- Continuous-time trajectory tracking can

An outline for my PhD

Completed Work: (Infrastructure Development)

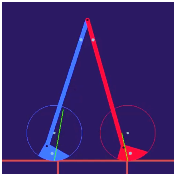
- Simulation for design and testing
- Trajectory optimization libraries
- Controller design and optimization algorithms

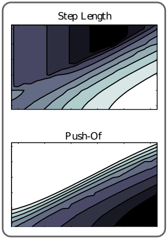
Planned Work:

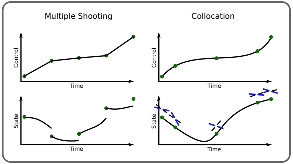
- Finalize controller framework (general robot)
- Test on high-fidelity Ranger simulation
- Implement on Ranger

Completed Work

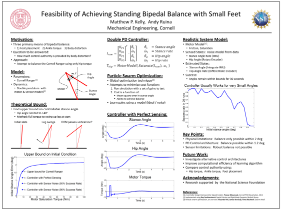
1. Background
2. My PhD Plan
3. Completed Work
    - Conference Presentations
    - Trajectory Optimization
    - Simulation Tools
    - Controller Design
4. Planned Work

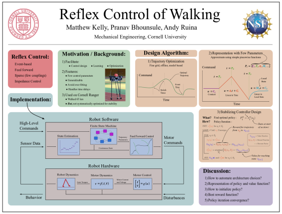


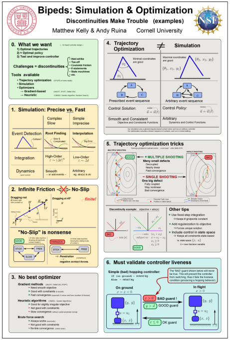




Poster Presentations at Dynamic Walking Conference 2012, 2013, 2014







Trajectory Optimization

Trajectory optimization

$\rightarrow$ Single solution to optimal control problem

Why is it useful?

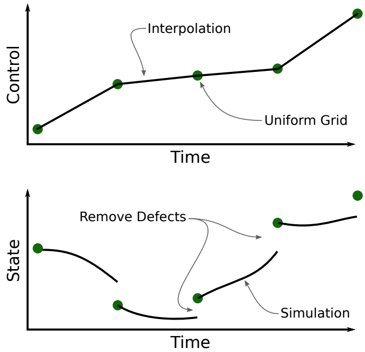
- Metric for comparison
- Providing insight about energy optimal behavior
- Can be used to create a trajectory library

How to solve? Transcription Methods:

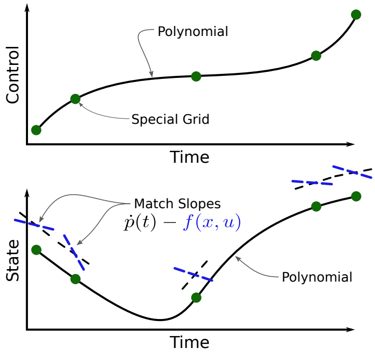
- Multiple Shooting
- Collocation

Transcription Methods

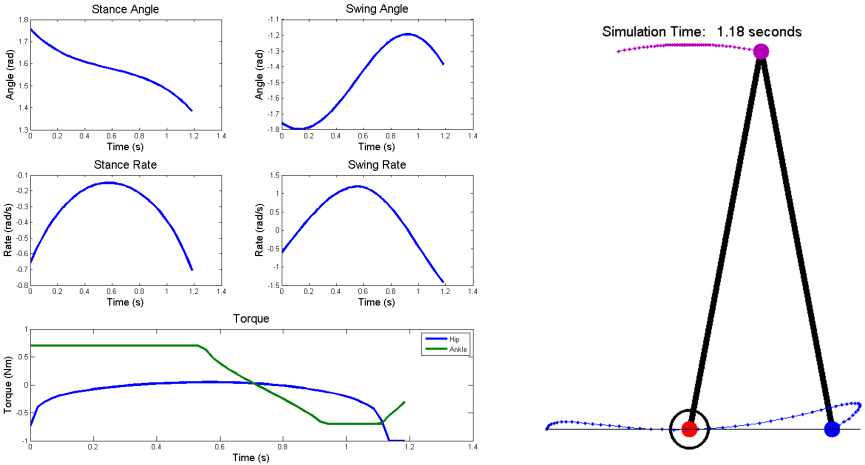
Multiple Shooting



Collocation



Trajectory Optimization - Example



Simulation Methods (for walking robots)

Event Detection

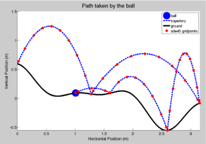
- Continuous dynamics
- Explicit finite state machine
- Arbitrarily accurate
- Requires simple system

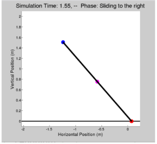
Time Stepping

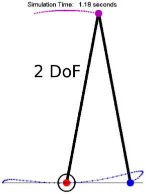
- Impulsive dynamics
- Implicit finite state machine
- Necessarily inaccurate
- Arbitrarily complex system

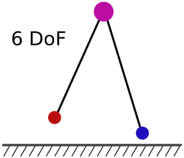
Simulated Systems

Event Detection

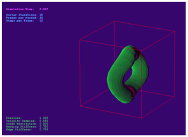


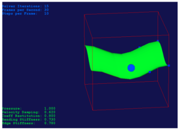


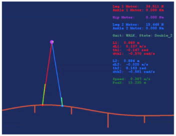


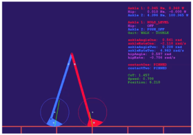


Time-Stepping









Point-Mass Controller Design

Optimal Solution

- Assume perfect model
- Find *control actions* such that:
  - Satisfy constraints
  - Minimize contraction ratio
- Each solution is independent
- Solver FMINCON

  Matlab non-linear constrained optimizer

Robust Controller

- Assume noisy actuation
- Find *controller parameters* such that:
  - Constraint Violation $\rightarrow 0$
  - Minimize weighted contraction ratio
- Entire solution is coupled
- Solver: CMAES

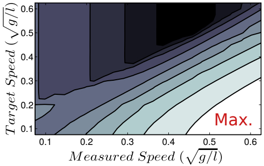
  Covariance Matrix Adaptation - Evolutionary Strategy

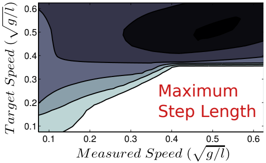
Point-Mass Controller Design

Perfect Model

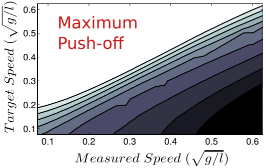
Noisy Model

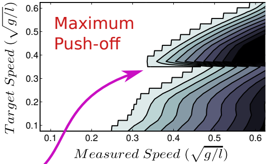
Step Length





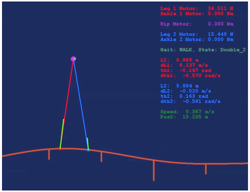
Push-off Impulse



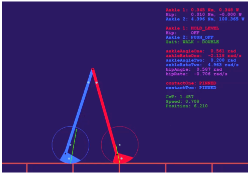


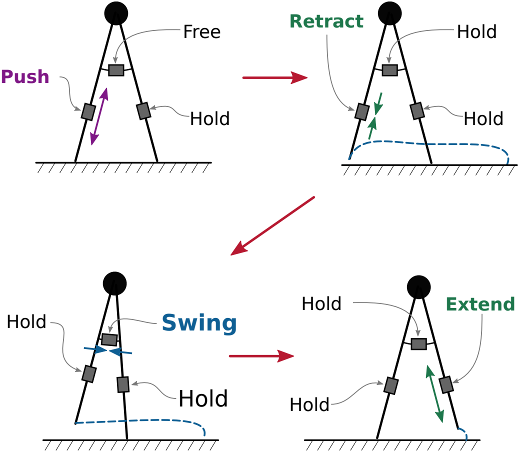
Interesting...

Continuous Controller Design



[Two Videos]

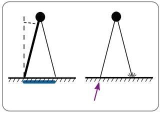




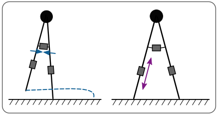
Planned Work

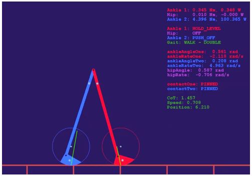
1. Background
2. My PhD Plan
3. Completed Work
4. Planned Work
   - Point-Mass Controller
   - Discrete → Continuous
   - Testing on Ranger
   - Deliverables

Reduced-Model Controller



Continuous Controller





Point-Mass Controller - Open Questions

Design:

- Best cost function?
- Best optimization architecture?

Model:

- Inputs? (target speed, ground slope, robustness, ...)
- How to model energy use?
- How to model actuator limits?
- Best noise model to use?

Continuous Controller Design - Open Questions

Double Stance

- Push-off impulse → equivalent torque profile?
- Energy considerations for push-off?

Single Stance

- Parametrization of swing foot trajectory?
- How to handle foot clearance and scuffing?
- How strong is coupling between swing foot and hip motion?

Simulation → Hardware: Testing on Ranger

Get I/O stuff working:

- Learn Ranger's control interface
- Create parallel control structure between simulation & reality

Test simple controller:

- Small motions in double stance
- Stepping in place

Walking controller:

- Start Moving from rest
- Walk at different speeds
- Come to a stop in double stance

Deliverables

Simulation of Ranger Walking

- Robust: Realistic sensor, actuation, and ground models
- Versatile: Speed: $0 \rightarrow v_1 \rightarrow v_2 \rightarrow 0$
- Efficient: Cost of Transport < 0.5

Put controller on Ranger

- It Works → How does it compare to simulation?
- It Fails → Why does it fail?

Publications:

- Conference Paper - Simulation results

  International Conference on Robotics and Automation (ICRA 2015)
- Journal Paper - Hardware results

  International Journal of Robotics Research

Acknowledgements

Comittee:

- **Andy Ruina**
- Ashutosh Saxena
- Hod Lipson
- Brandon Hencey

Ruina Lab:

- Petr, Anoop, Matt, Boris, Atif, and Jason

Funding:

- National Science Foundation

Logistics:

- Marcia Sawyer

Feedback:

- Hadas Kress-Gazit, Tim Lannin

Questions?

Equations of Motion for Point-Mass Walker

Swing down to collision

$$\dot{\theta}^- = \sqrt{\dot{\theta}_k^2 + \frac{2g}{l}(1 - \cos\phi)}$$

Push-off followed by heel-strike

$$\dot{\theta}^+ = \dot{\theta}^- \left(\cos^2\phi - \sin^2\phi\right) + \frac{2p}{ml}\cos\phi\sin\phi$$

Swing back up to mid-stance

$$\dot{\theta}_{k+1} = \sqrt{(\dot{\theta}^+)^2 - \frac{2g}{l}(1 - \cos\phi)}$$

Constraint Equations for Point-Mass Walker

Prevent flight before, during, and after collision

$$c < 0 \qquad c = (2ml)\dot{\theta}^- \cos\phi \sin\phi - p\left(\cos^2\phi - \sin^2\phi\right)$$

$$\dot{\theta}^- < \dot{\theta}_{fly} \qquad \dot{\theta}_{fly} = \sqrt{\frac{g}{l}\cos\phi}$$

$$\dot{\theta}^+ < \dot{\theta}_{fly}$$

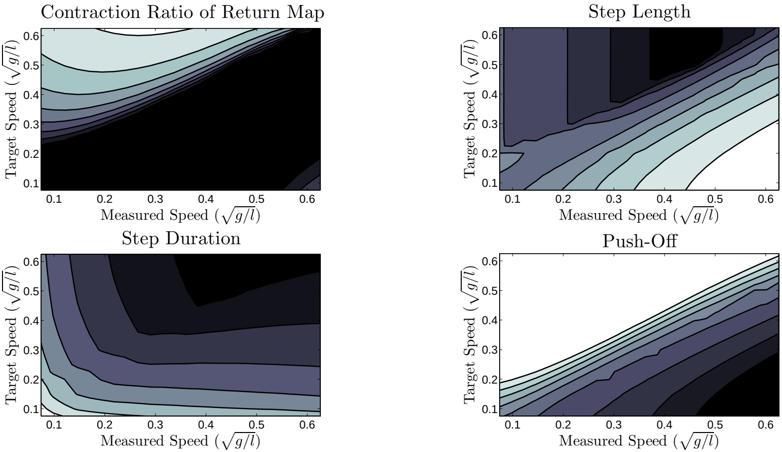
Prevent falling backwards during swing-up

$$\dot{\theta}^2_{fall} > 0 \qquad \dot{\theta}^2_{fall} = (\dot{\theta}^+)^2 - \frac{2g}{l}(1 - \cos\phi)$$

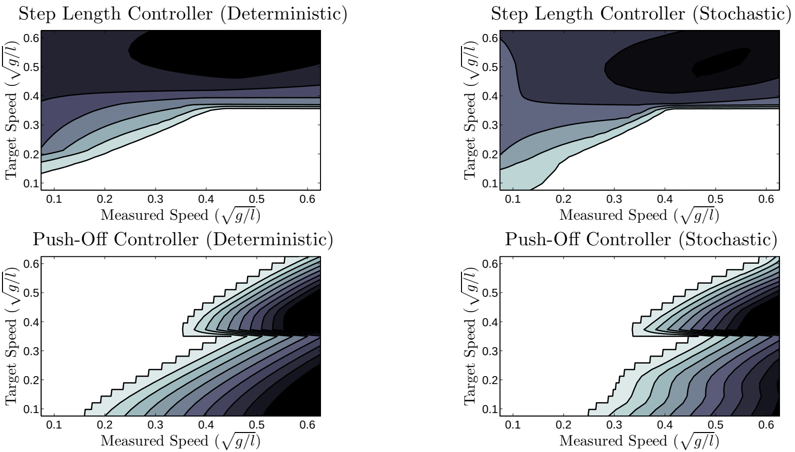
Allow time for leg swing

$$t_{step} > t_{min} \qquad t_{step} = \frac{(\phi_{k+1} - \phi_k)^2}{\int_{\phi_k}^{\phi_{k+1}} \dot{\theta} d\theta}$$

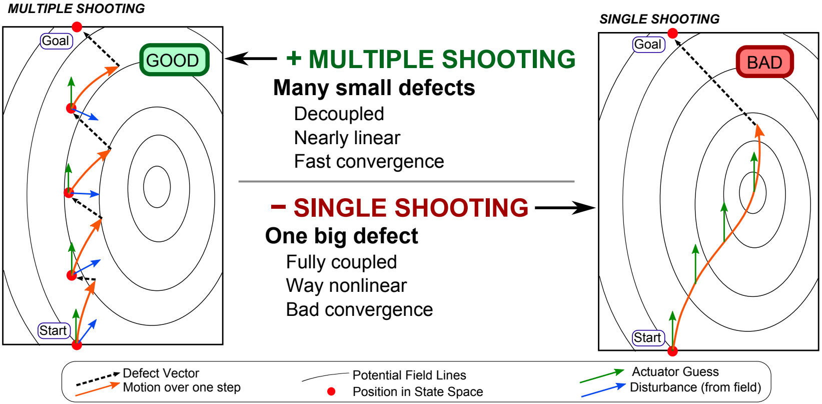
Deterministic Optimal Control for Point-Mass Walker



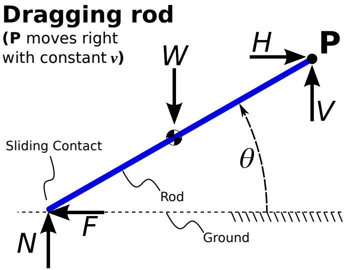
Robust Control for Point-Mass Walker

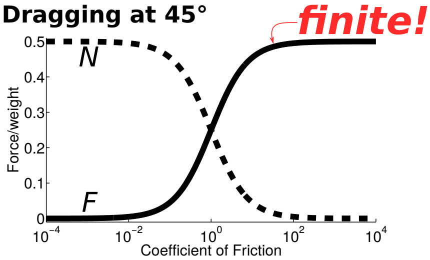


Multiple Shooting versus Single Shooting



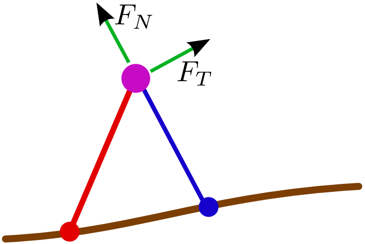
Contact Mechanics – Infinite Friction





Biped Simulator - Control Algorithm

Double Stance



Length Controller:

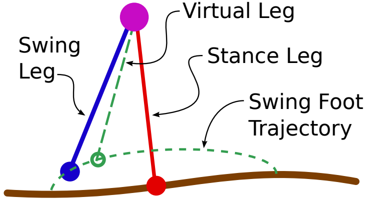
$$F_N = k_p(\dot{l}_{target} - l_1) - k_d \dot{l}_1$$

Energy Regulator:

$$\dot{\theta}_{target} = \frac{-1}{l_1}\sqrt{v_{target}^2 + 2gl_1\cos(\theta_1)}$$

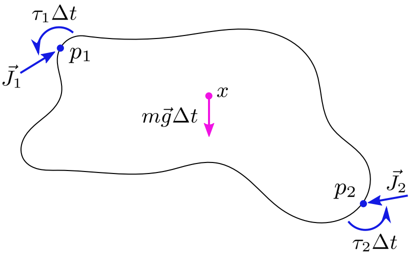
$$F_T = \frac{-k}{l_1}(\dot{\theta}_{target} - \dot{\theta})_1$$

Single Stance



- Bezier curve
- Phase variable
- Pose control
- PD tracking

Ranger Simulator - Dynamics Equations



Compute center of mass kinematics:

$$\Upsilon = \hat{\mathbf{k}} \cdot \left((p_1^- - x^-) \times \vec{J}_1 + (p_2^- - x^-) \times \vec{J}_2 \right) / I$$

$$\dot{\theta}^+ = \dot{\theta}^- + \tau_2\Delta t + \tau_1\Delta t + \Upsilon$$

$$\dot{x}^+ = \dot{x}^- + \left(\vec{J}_1 + \vec{J}_2 + m\vec{g}\Delta t \right) / m$$

Compute contact point kinematics:

$$\theta^+ = \theta^- + \dot{\theta}^+\Delta t$$

$$x^+ = x^- + \dot{x}^+\Delta t$$

$$p_1^+ = p_1^- + \Delta t \left(\dot{x} + (\dot{\theta}^+\hat{\mathbf{k}}) \times (p_1^- - x^-) \right)$$

$$p_2^+ = p_2^- + \Delta t \left(\dot{x} + (\dot{\theta}^+\hat{\mathbf{k}}) \times (p_2^- - x^-) \right)$$

Pin-joint constraint:

$$0 = p_A^+ - p_B^+$$

Free-joint constraint:

$$0 = \vec{J}_A$$