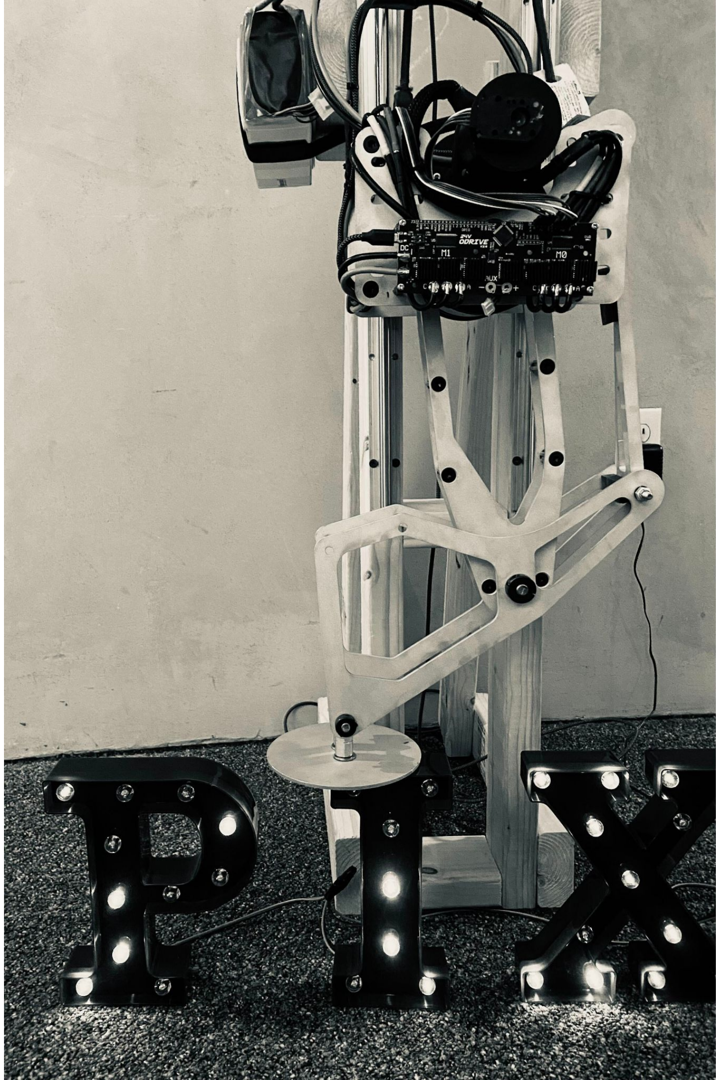
PIXAR

luxo jr. (the pixar lamp)

Analyzing the effects of soft ground contact on the ability of a real system to reach a steady-state jump height.

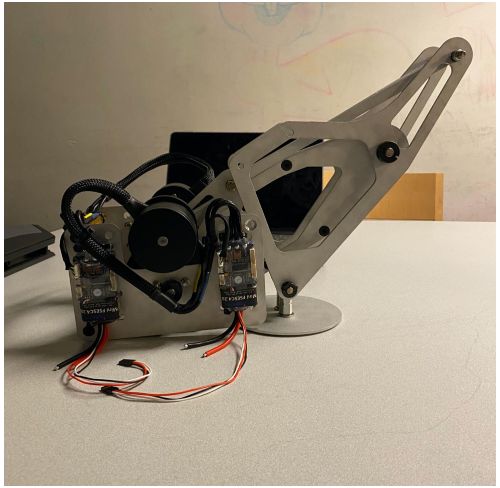
Adi Mehrotra, Fischer Moseley, Alexander Tsao



Introduction

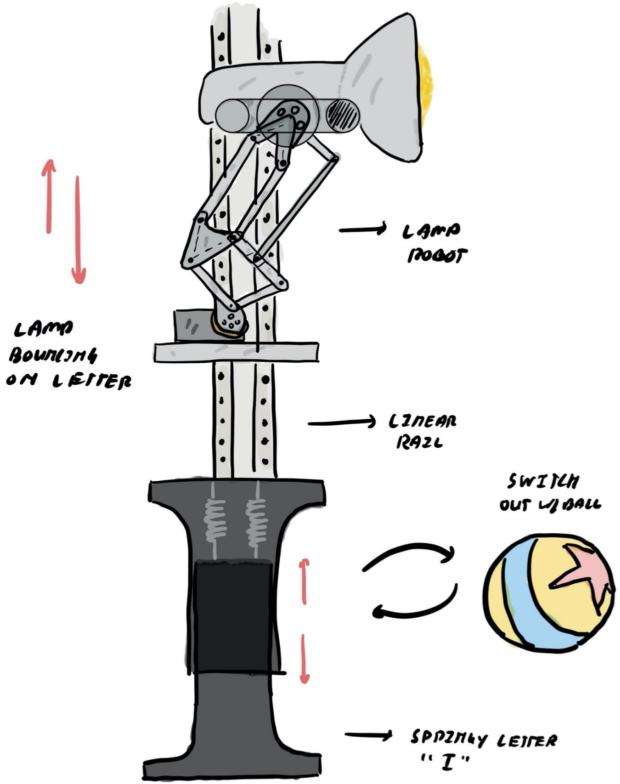
The Pixar lamp jumps on an elastic “I”

Goal: Replicate this soft ground contact and optimize a controller to achieve a stable jump height oscillation.

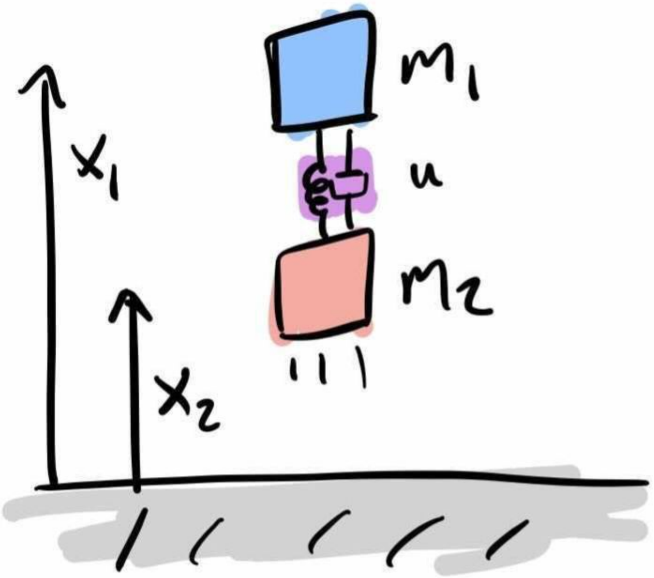


Basic System Model

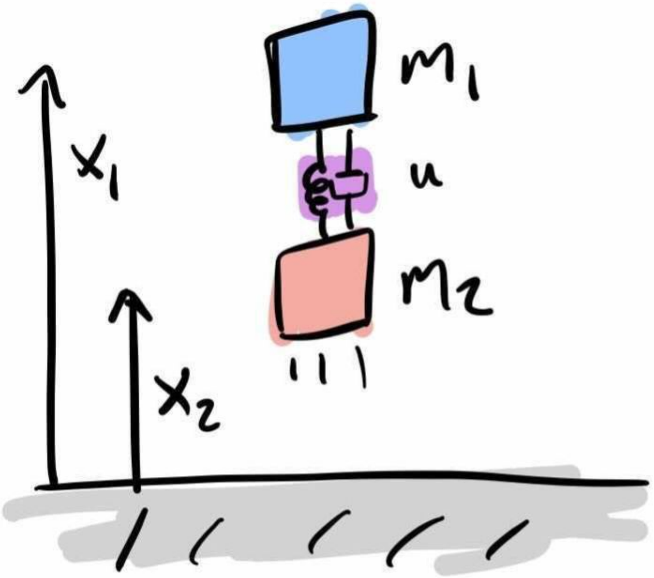
- 2-DOF leg
- hip is constrained to a vertical slider
- foot is unconstrained



Simulation Methods



Simulation Methods

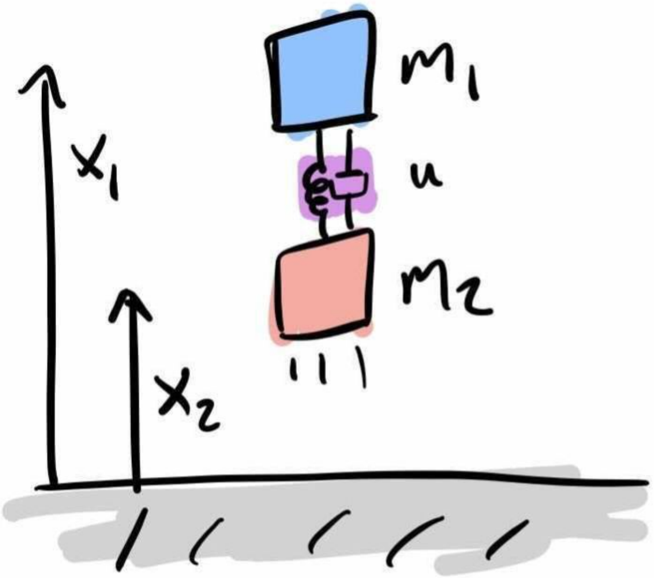


$$m_1 \ddot{x}_1 = -m_1 g + u$$

$$m_2 \ddot{x}_2 = -m_2 g - u + F_g$$

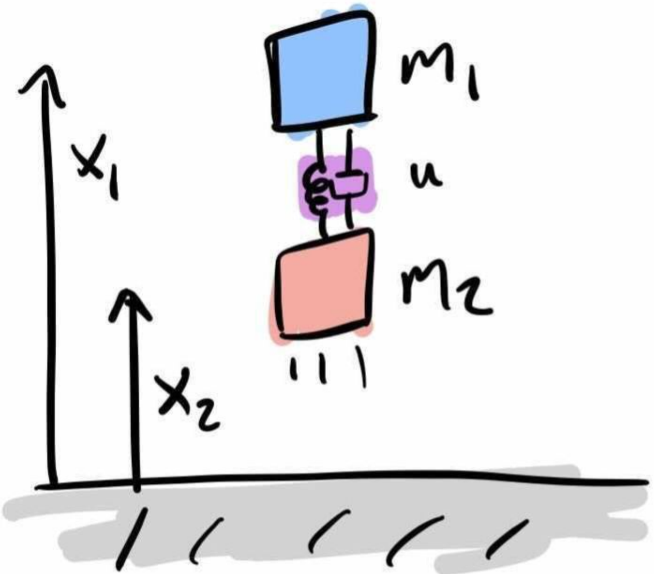
$$F_g = \underbrace{\left(\frac{1}{\pi}\arctan(\varepsilon x_2) + \frac{1}{2}\right)}_{\text{smoothing term}} \cdot \left(-k x_2 - d\dot{x}_2\right)$$

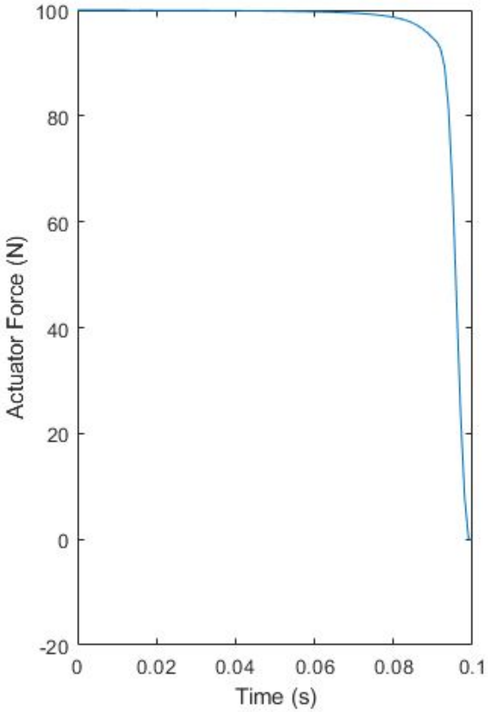
Simulation Methods

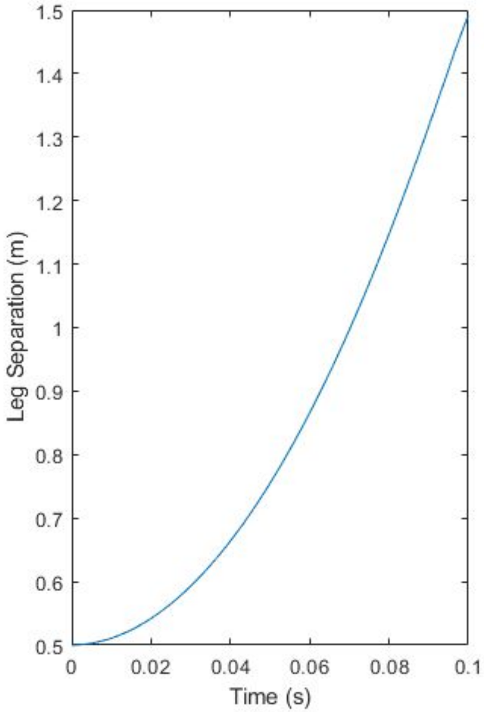


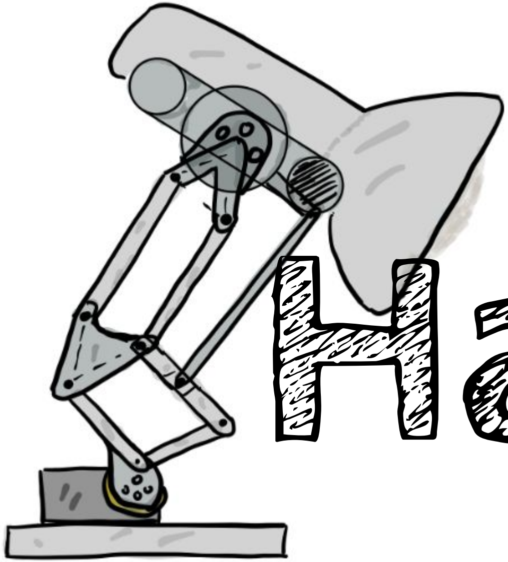
- Direct shooting optimization (IPOPT)
- Decision variables: duration and trajectory of control
- Subject to:
    - Bounding box constraint on separation between hip and foot
    - Actuator limit
    - Springy ground dynamics

Simulation Results









Hardware

Hardware

(is hard...)

General Overview

Mechanical:

- Linear rail and carriage
- Waterjet aluminum frame
- ~~Spring-loaded 'I'~~

→ **very heavy hip, used a bungee**

Actuators:

- 2x brushless motors

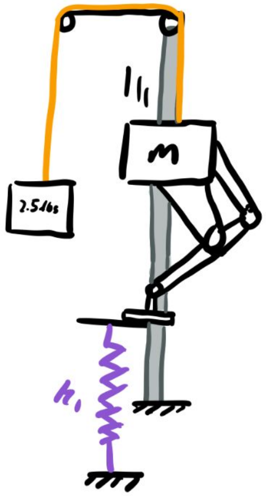
Control:

- ~~Teensy microcontroller~~
- ~~VESCs~~

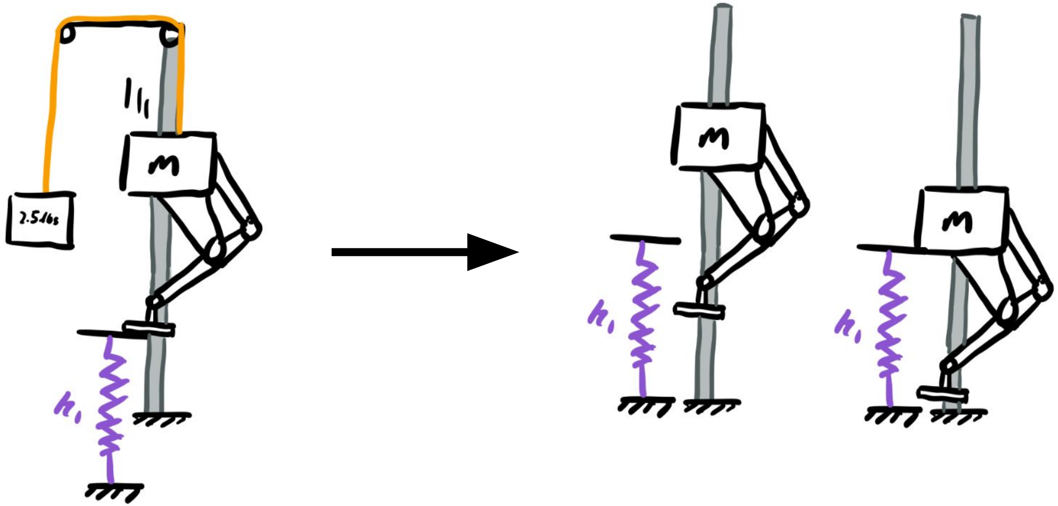
too much torque ripple, added encoders for FOC

Used an ODrive instead!

Counterweight vs Bungee

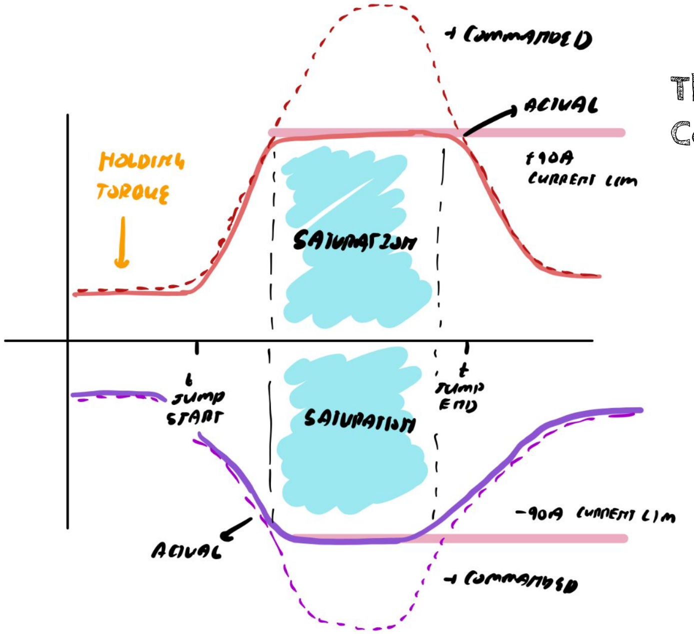


Counterweight vs Bungee



(bungee cords in orange)

Initial Control Strategy



The "Full-Send" Controller

TYPICAL IMPEDANCE CONTROLLER

$$\tau \propto k[\theta_e] + d[\dot{\theta}_e]$$

Impedance Controller:

- Basic impedance controller
- Used this to track a recorded trajectory

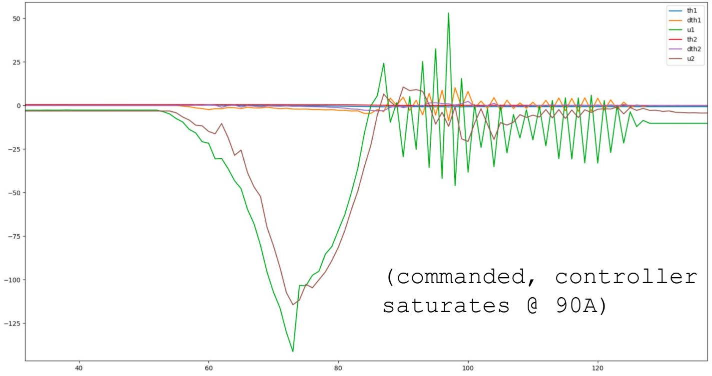
Process:

- Recorded a "jump" trajectory with desired positions and velocities
- Tried to maximize joint torques to jump as high as possible

Intermediate Results

- 2.5 lbs counterweight
- controller attempts to optimize for jump height

It learned bang-bang control!!!, like the sim showed!!!



Control Strategy - Energy Pumping

DROP

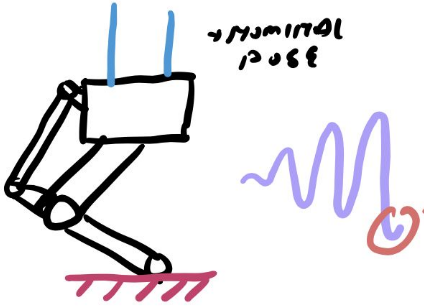
$$mgh = \tfrac{1}{2}kx^2 + \underbrace{E_{FOOT}}_{\text{ENERGY ABSORBED}} + \underbrace{E_{IMPACT}}_{\text{ENERGY LOST TO IMPACT}}$$

SEND

$$mgh = \tfrac{1}{2}kx^2 + E_{FOOT}$$

$$mgh = E_{FOOT} + E_{IMPACT}$$

- mgh: MEASURE W/ SLOW-MO CAMERA
- E_{FOOT}: CSV DATA FROM MOTORS & SUCH
- E_{IMPACT}: DIFF IS LOST TO ENV



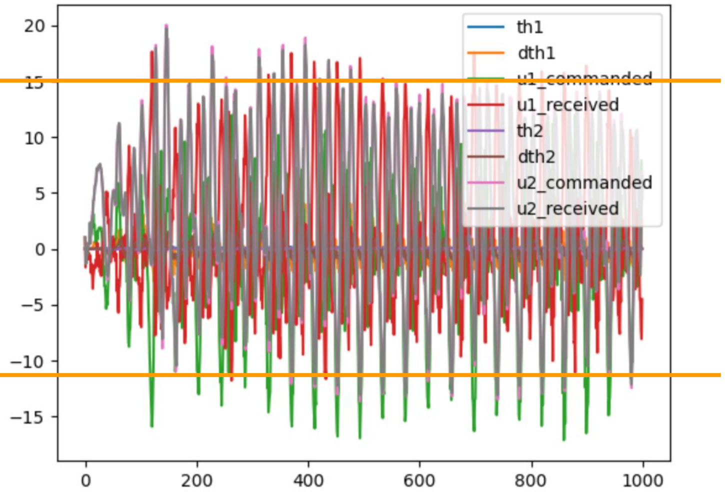
+ NOMINAL POSE

+ BOUNCE RAMP UP & DOWN & WATCH CURRENT RISE

THEN DO A FULL SEND FROM HERE

Final Results - IT JUMPS!!!!

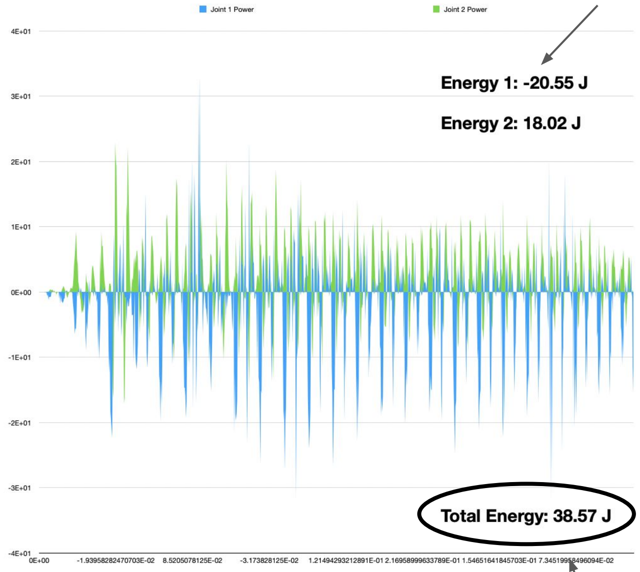
Steady state reached

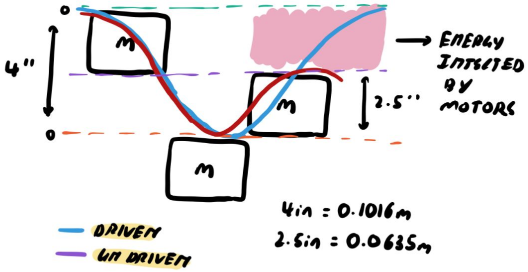


So basically, it's putting any energy lost back into the spring each cycle... let's prove that!

Final Results - Energy Analysis

(-) just a coordinate flip...





4in = 0.1016m
2.5in = 0.0635m

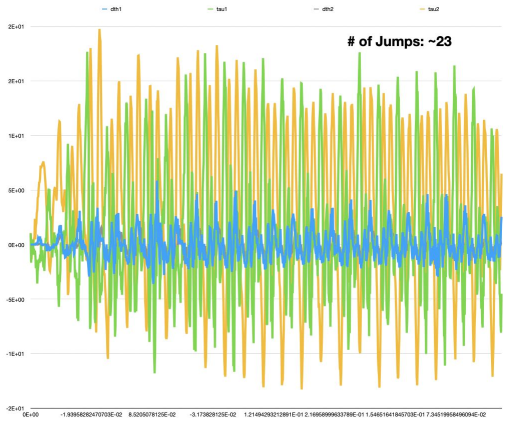
$m \approx 5kgs \quad g \approx 9.8\ m/s^2$

$\Delta E = 5\ kgs \cdot (9.8 m/s^2) \cdot (0.1016 m - 0.0635 m)$

$\Delta E = 1.8669 J \cdot 23 = 42.92387 J$

23 CYCLES

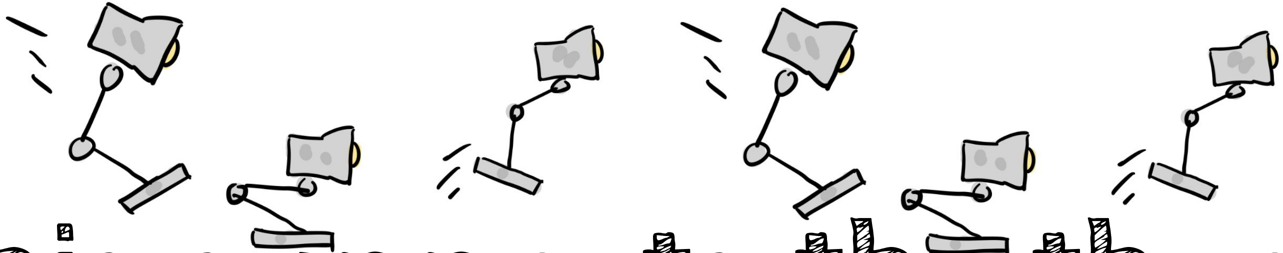
only 4 J difference!



Possible Discrepancies:

- We **assumed 23 complete jump cycles** but the first few were "pumping" and the last was "deceleration"
- **assumed energy was the same each cycle**

**mass est. not measured,
+ got heights by recreating video by hand



This supports the theory…

On its own, the lamp would lose energy.

The motors pump any lost energy back into the system to maintain a stable limit cycle.

Takeaways

- Hall effect sensors give poor position estimates at low speed
- The unconstrained foot was a disaster
  - Added a spring to provide a rest position
- Key Takeaway:
  - Intuitively, the leg must inject enough energy into the spring-mass system to maintain stable oscillation
  - But to write an optimizer we'd need to know the spring position, contact detection, CoM position…
    - all difficult to measure in real life

References

- MIT 6.832/843 notes
- Matt Chignoli's HW5 optimization code
- ODrive references
- Thanks to:
  - Professor Russ Tedrake for advice on trajectory optimization
  - Elijah for teaching Adi about motors
  - Andrew for advice on research question
  - All the course staff + Professor Sangbae Kim
- Special thanks to Pixar for letting us completely violate copyright law

Come talk to us after!

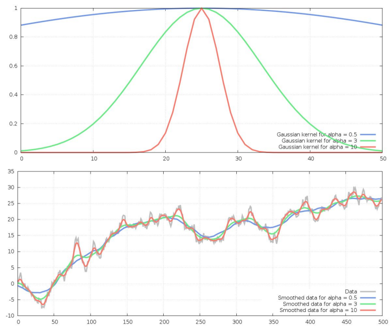
Integration Pains (Appendix)

VESC Problems:

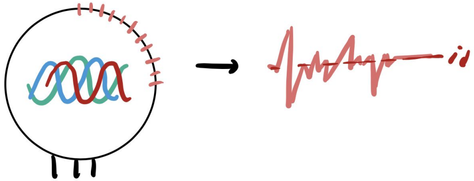
- VESC motors do 6-step commutation
- This causes large amounts of torque ripple
- Which shows up as velocity "noise"

Key Takeaway:

- Made it incredibly difficult to tune a stable impedance controller



(example from google...)



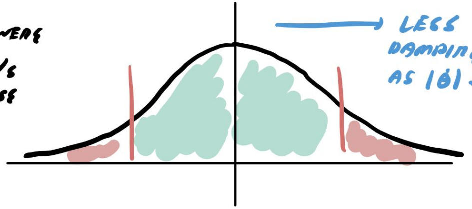
$\tau = k(\theta_d - \theta_c) + d(\dot{\theta}_d - \dot{\theta}_c)$

Now there's two problems w/ this

Gaussian dist. $G(\dot{\theta}_e) = e^{\frac{\dot{\theta}_e^2}{\sigma^2}}$ $\quad \dot{\theta}_e = \dot{\theta}_d - \dot{\theta}_c$

$\therefore \quad \tau = k\theta_e + d \cdot e^{\frac{\dot{\theta}_e^2}{\sigma^2}}$

+ spikes were ~40 rads/s so we chose $\sigma = 10$



less damping as $|\dot{\theta}|$ inc.

**moving average filter doesn't work b/c this filters out useful data we need to control (not noise)

Came up w/ a control strategy that uses gaussian filters on velocity term