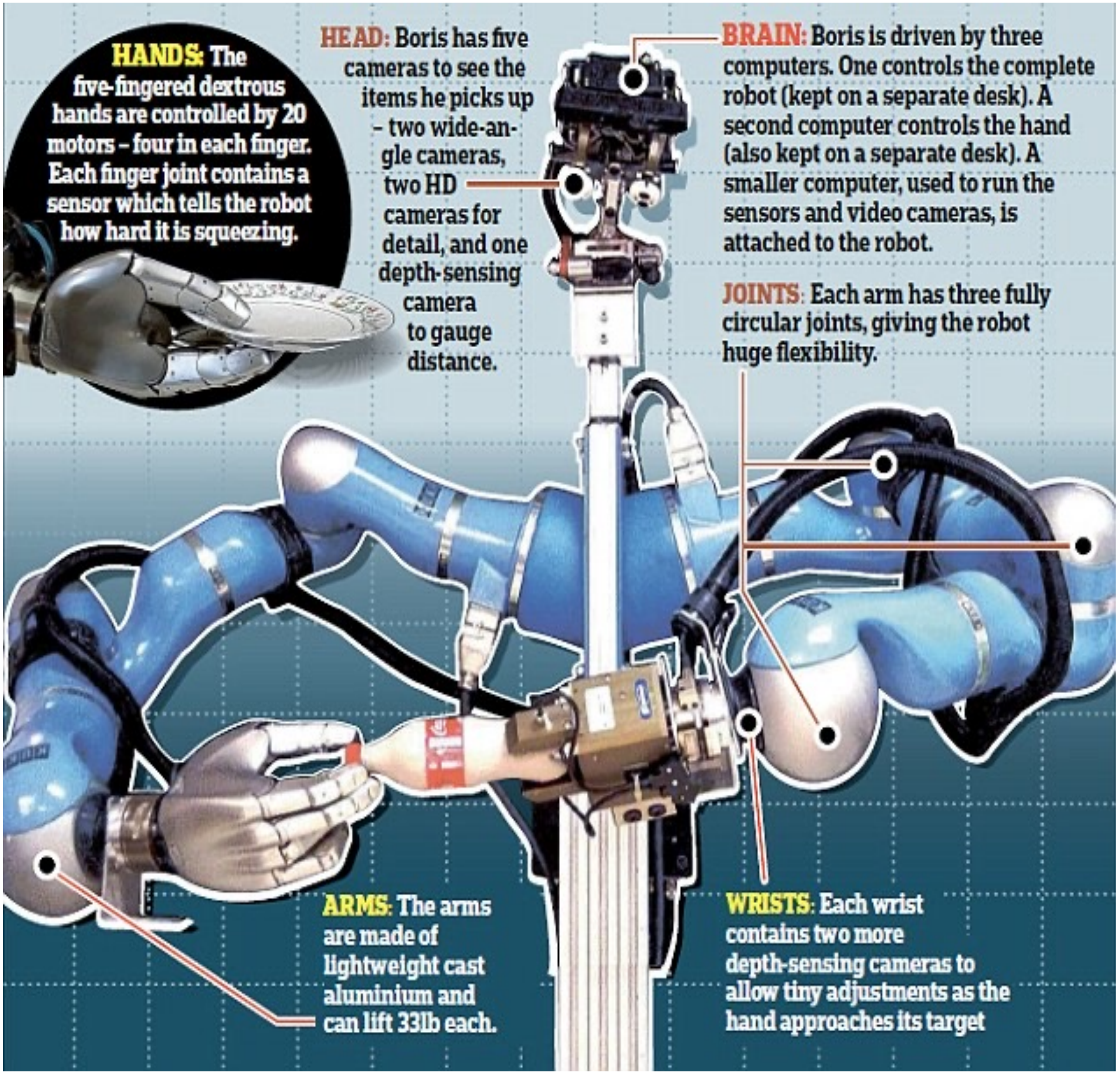
Automated Design of Special Purpose Dexterous Manipulators

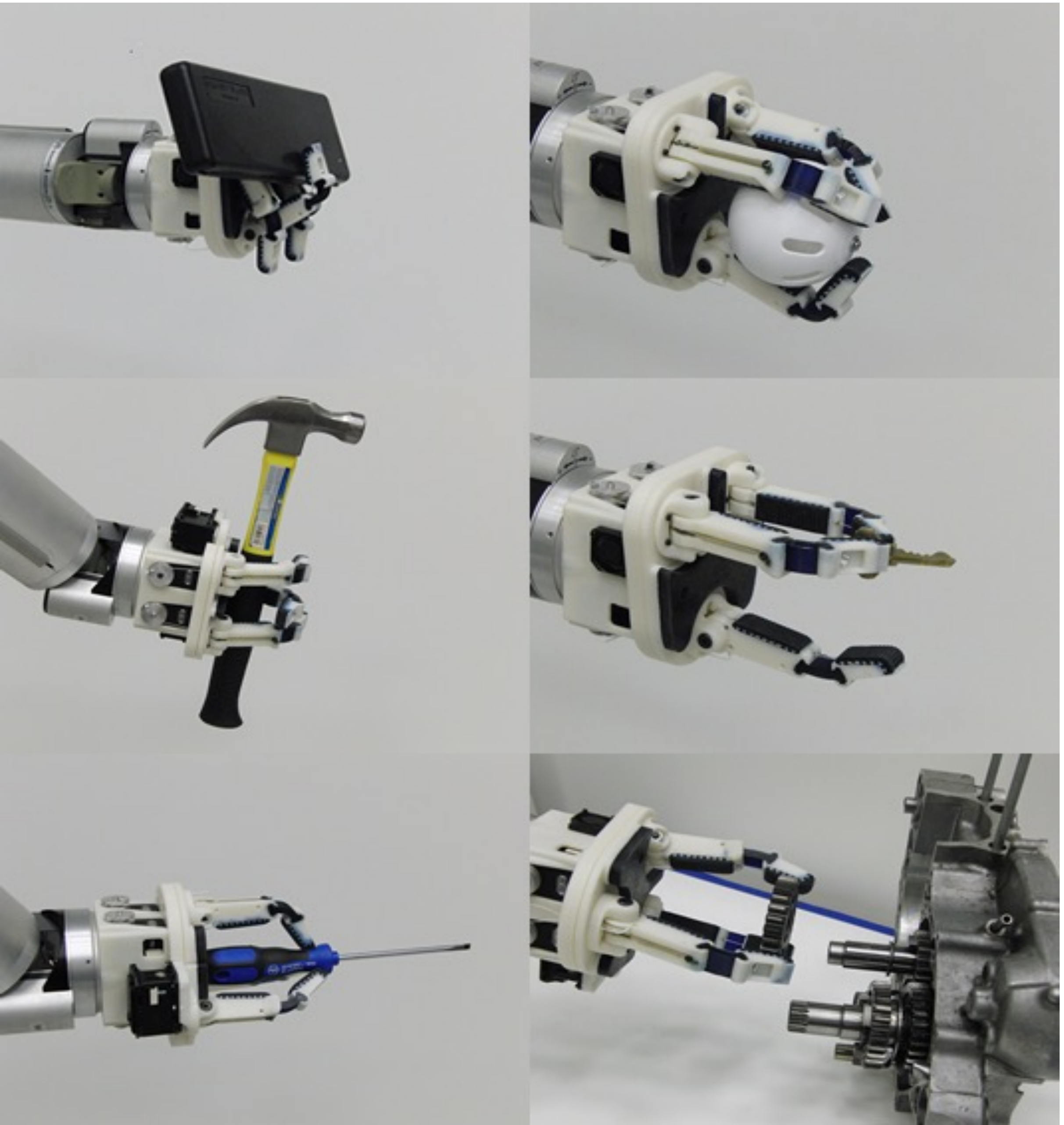
Christopher Hazard

Motivation


HANDS: The five-fingered dextrous hands are controlled by 20 motors – four in each finger. Each finger joint contains a sensor which tells the robot how hard it is squeezing.
HEAD: Boris has five cameras to see the items he picks up – two wide-angle cameras, two HD cameras for detail, and one depth-sensing camera to gauge distance.
BRAIN: Boris is driven by three computers. One controls the complete robot (kept on a separate desk). A second computer controls the hand (also kept on a separate desk). A smaller computer, used to run the sensors and video cameras, is attached to the robot.
JOINTS: Each arm has three fully circular joints, giving the robot huge flexibility.
ARMS: The arms are made of lightweight cast aluminium and can lift 33lb each.
WRISTS: Each wrist contains two more depth-sensing cameras to allow tiny adjustments as the hand approaches its target

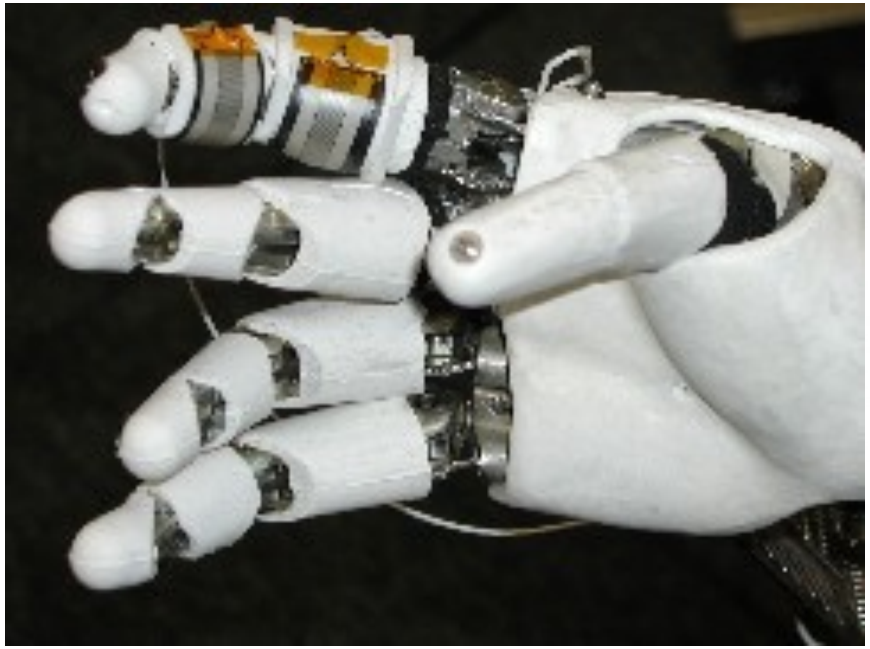

http://robot.watch.impress.co.jp/

Challenges



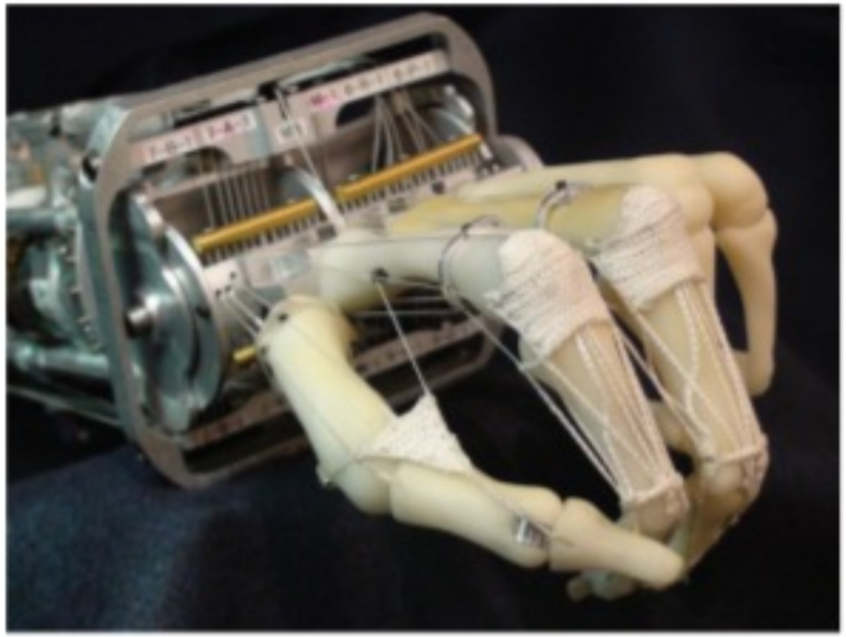
Humanoid Hands

- Goal: mirror human hand
- Impressive capability
- Important limitations
- Very expensive
- Costly mechanical failures



NASA Robonaut Hand

Shadow Dexterous



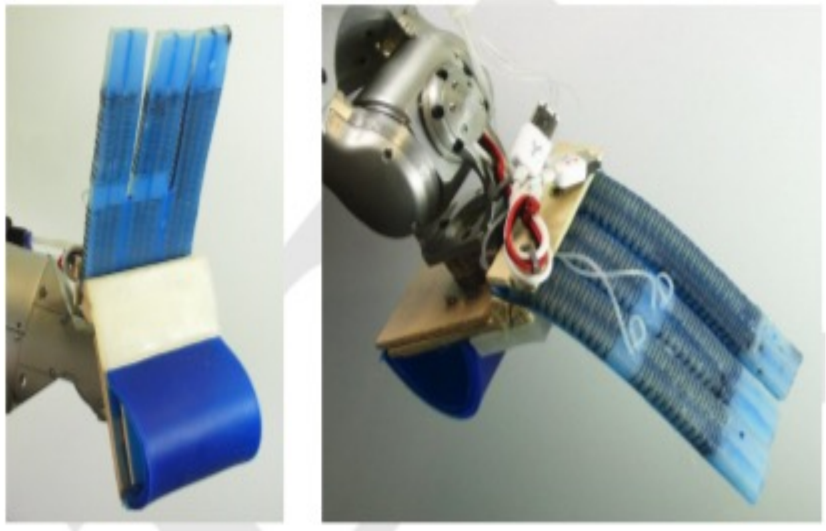
ACT: anatomically correct testbed hand

[1] “The ACT hand: Design of the skeletal structure” Weghe 2004

[2] “The robonaut hand: A dexterous robot hand for space” Lovchick 1999

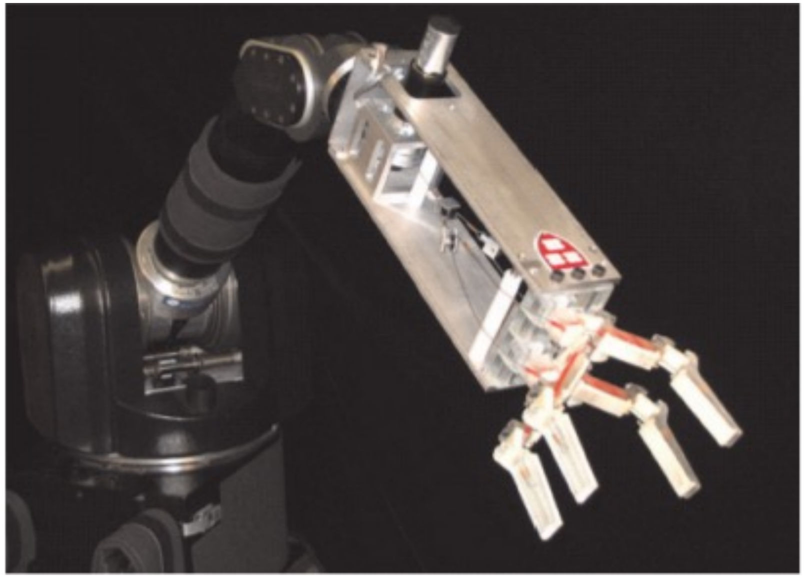
[3] Shadow Dexterous: https://www.shadowrobot.com

Low Cost (Simplified) Hands



Pneumatic Hand (Diemel 2013)

3D printed Hand (Ma 2013)

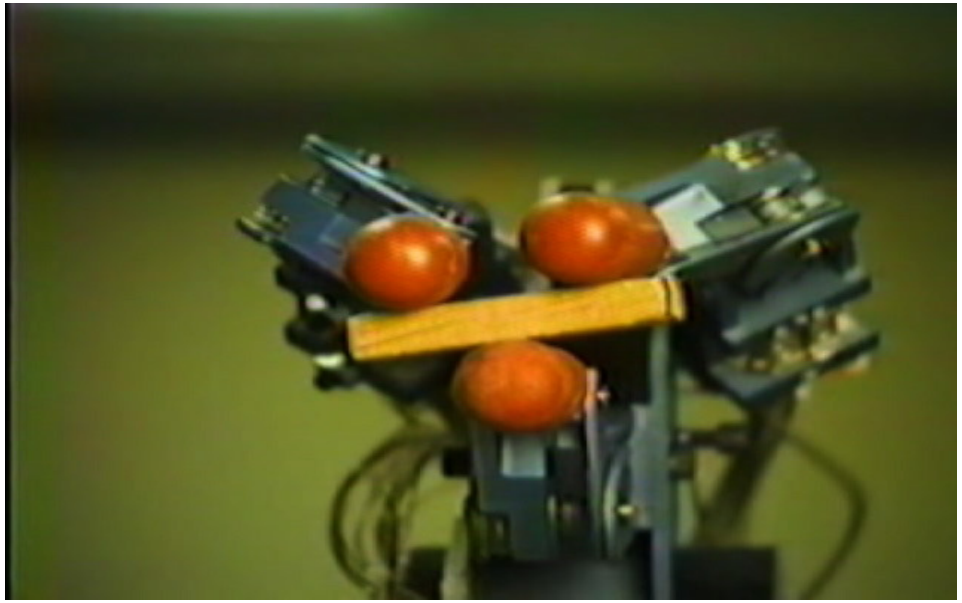


SDM Hand (Dollar 2010)

- underactuated designs
- 3d printable components
- cheap materials + simple construction
- soft/compliant components
- cheap embedded sensing

[1] “A modular, open-source 3D printed underactuated hand” Ma 2013
[2] “A compliant hand based on a novel pneumatic actuator” Deimel 2013
[3] "The highly adaptive SDM hand: Design and performance evaluation“ Dollar 2010

Design Parameter Optimization



Salisbury 1982: Stanford-JPL hand

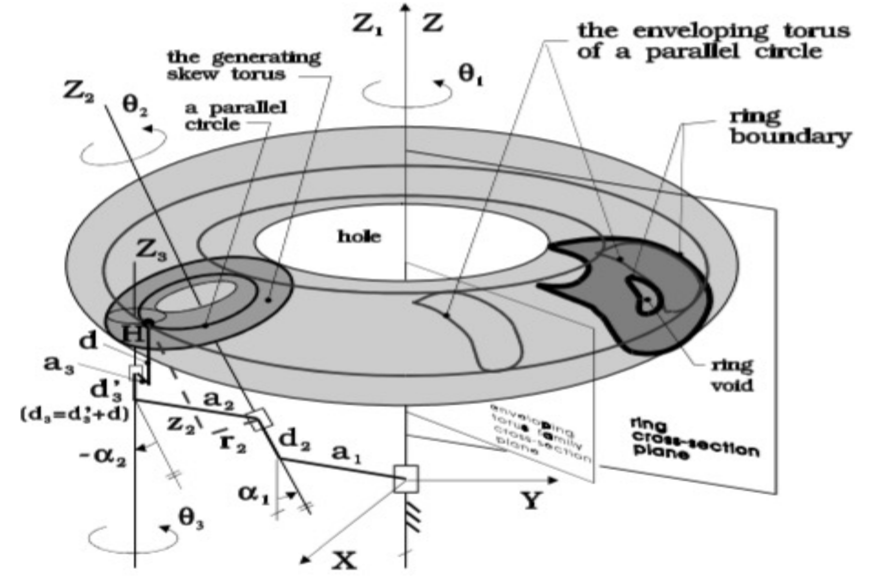
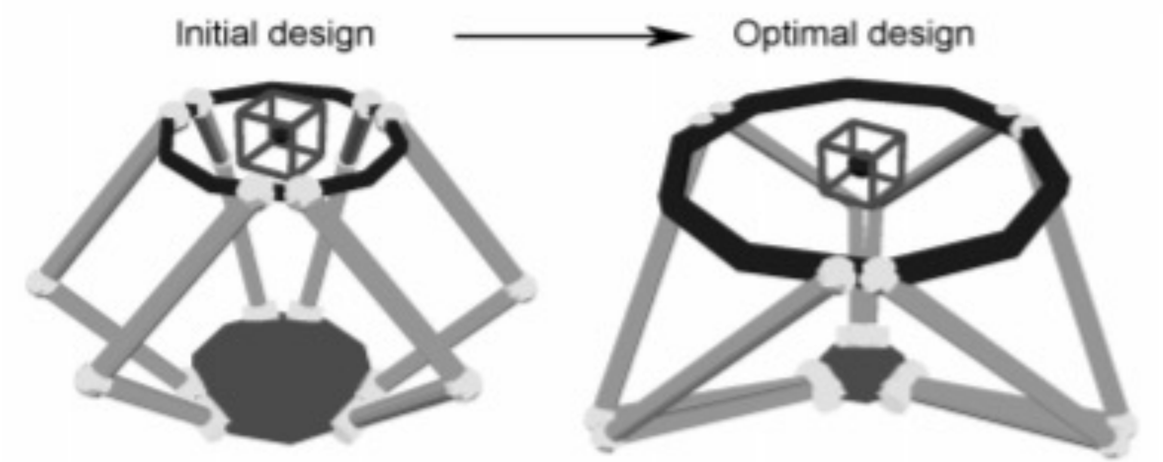


Fig. 2. Design parameters and workspace geometry for 3R manipulators.

Ceccarelli 2004: workspace optimization



Collard 2005: Manipulability Optimization

[1] “Articulated hands: Force control and kinematic issues” Salisbury 1982

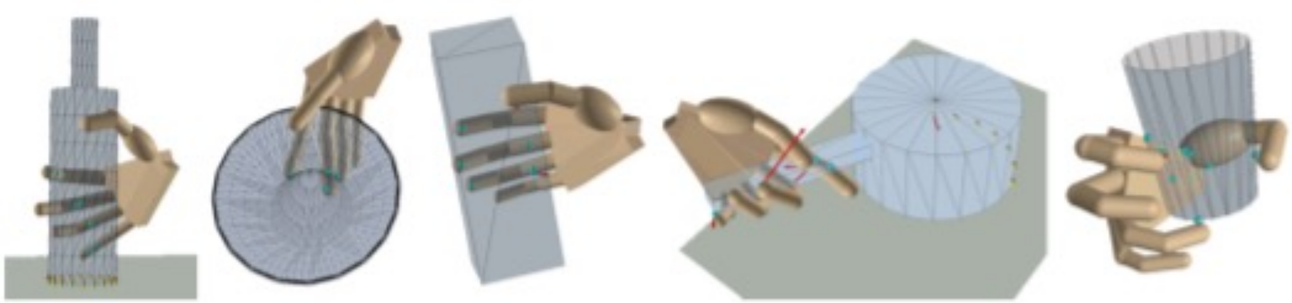
[2] “A multi-objective optimum design of general 3R manipulators for prescribed workspace limits” Ceccarelli 2004

[3] “Contribution to the optimization of closed-loop multibody systems: Application to parallel manipulators” Collard 2005

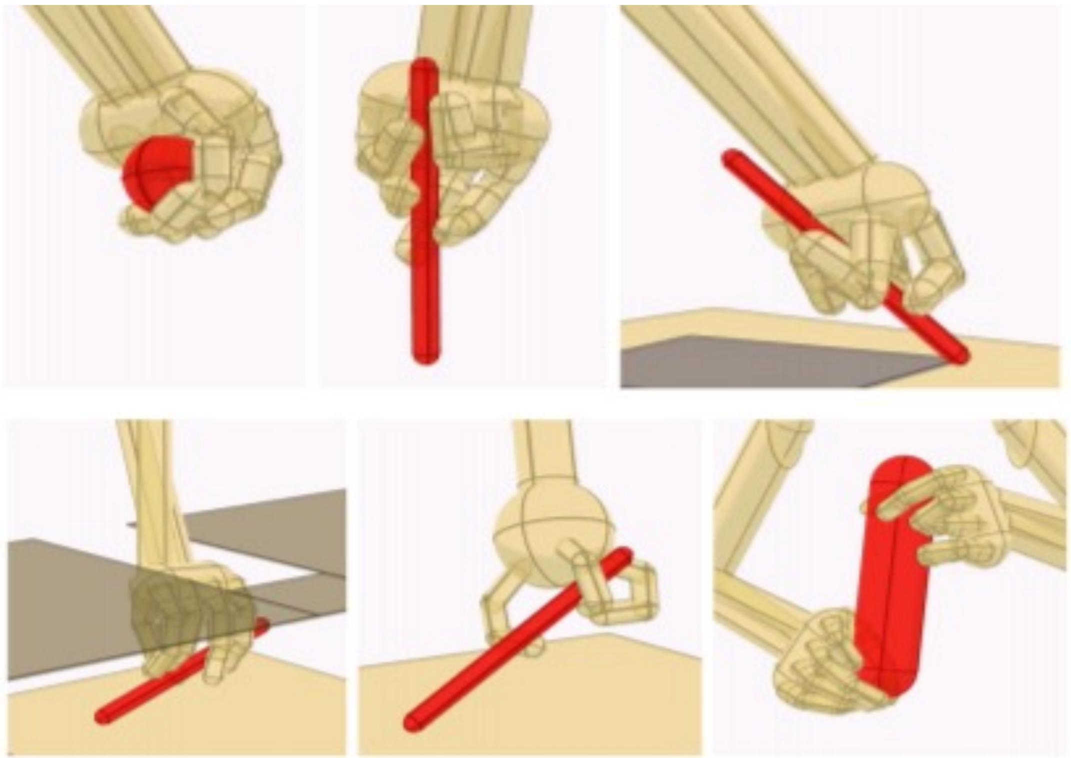
[4]” An optimization problem approach for designing both serial and parallel manipulators” Ceccarelli 2005

Trajectory Optimization

Liu 2008



Ye 2012



Mordatch 2012

[1] “Construction and animation of anatomically based human hand models” Albrecht 2003

[2] "Synthesis of interactive hand manipulation." Liu 2008

[3] ”Dextrous manipulation from a grasping pose” liu 2009

[4] “Synthesis of Detailed Hand Manipulations Using Contact Sampling” Ye 2012

[5] "Contact-invariant optimization for hand manipulation." Mordatch 2012

Our Work

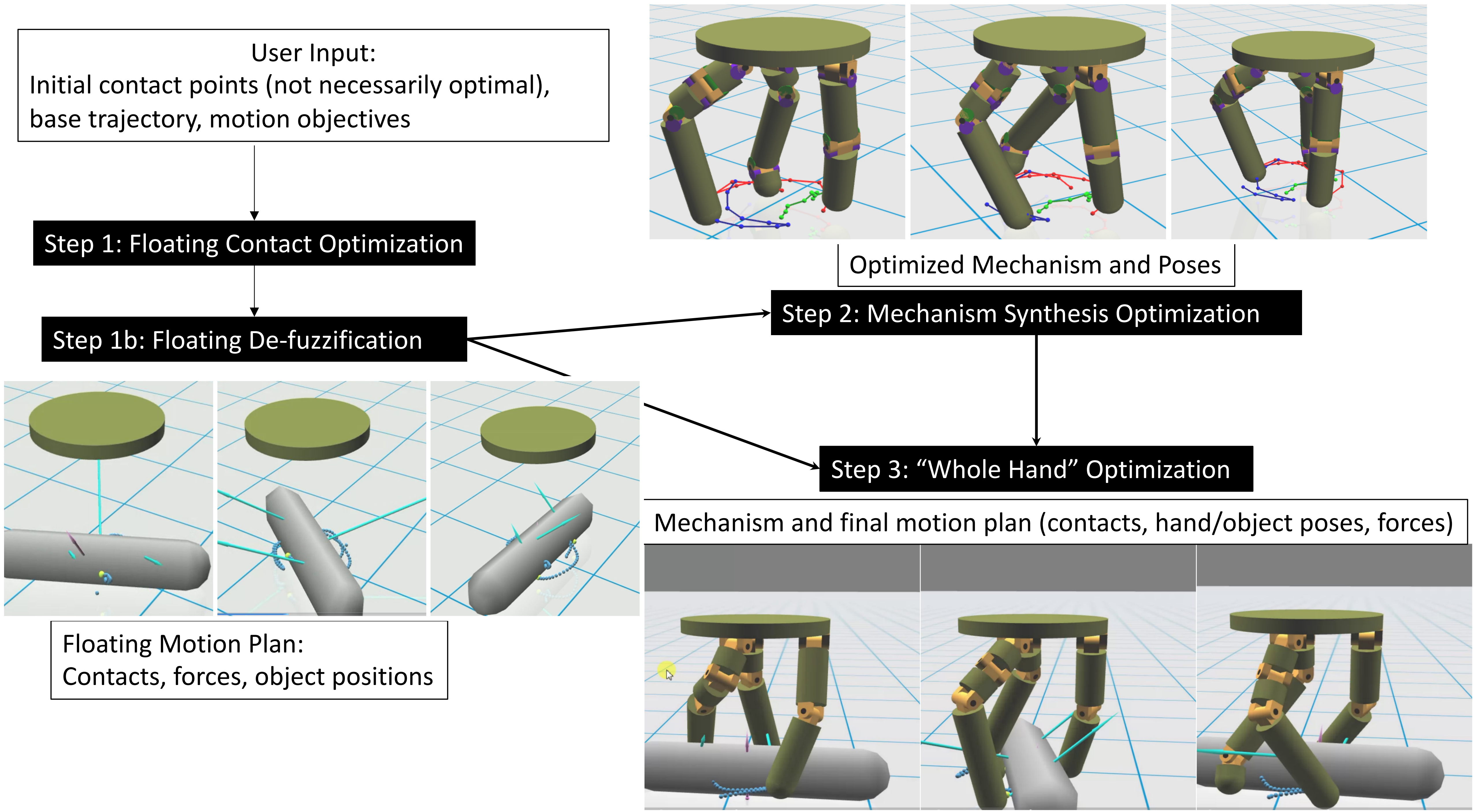
High Level Task Input

↓

Optimization

↓

Simplified Hand Design + Motion Plan

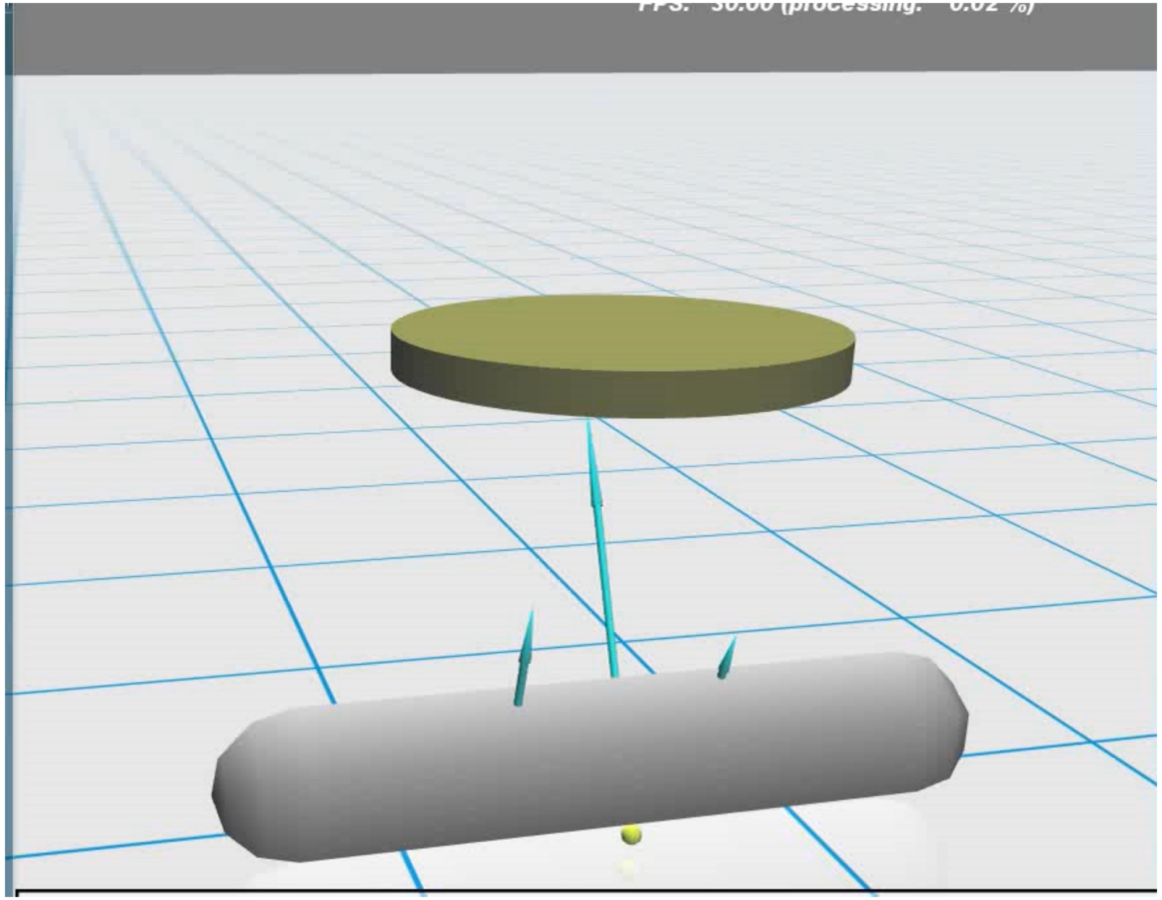

User Input:
Initial contact points (not necessarily optimal), base trajectory, motion objectives
Step 1: Floating Contact Optimization
Step 1b: Floating De-fuzzification
Floating Motion Plan:
Contacts, forces, object positions
Optimized Mechanism and Poses
Step 2: Mechanism Synthesis Optimization
Step 3: “Whole Hand” Optimization
Mechanism and final motion plan (contacts, hand/object poses, forces)

Step 1: Floating Contact Optimization

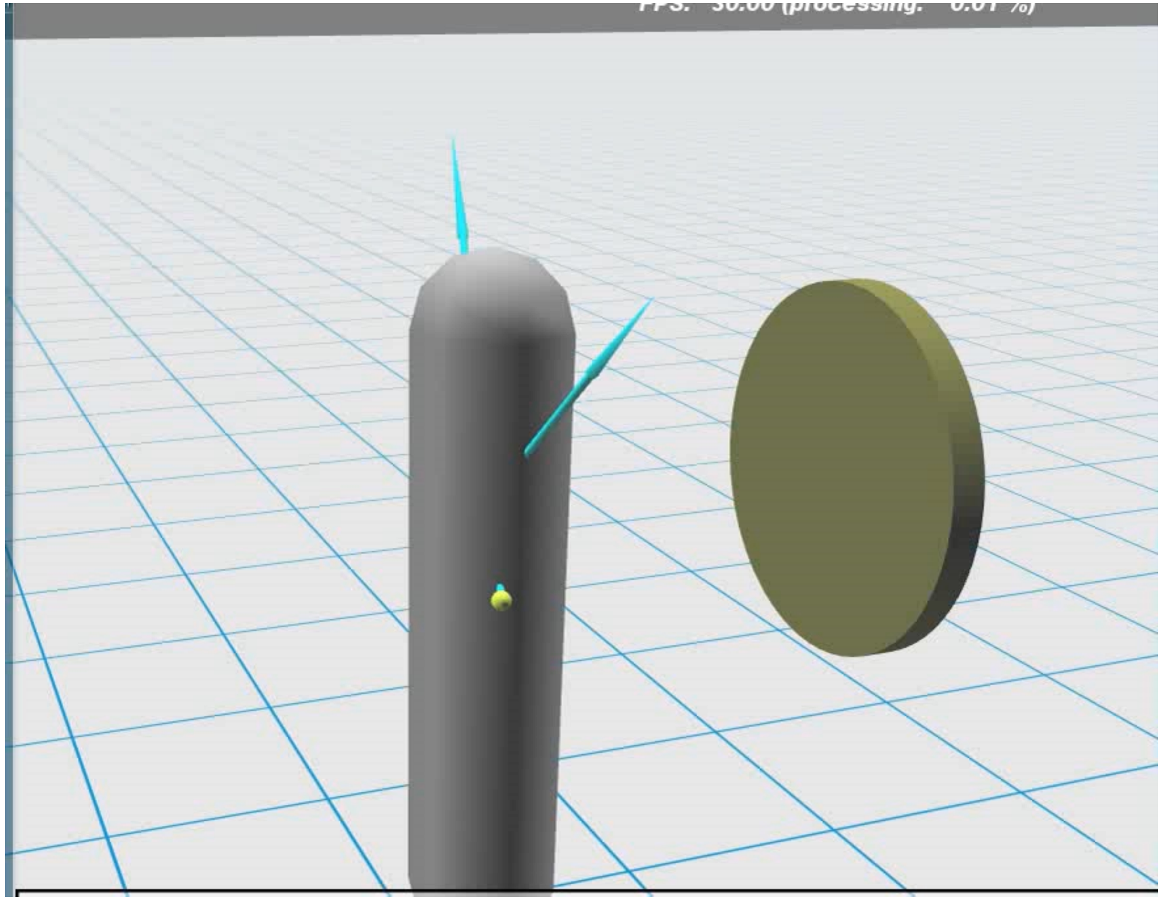
Floating Contact Optimization

Input:
- Object goal poses
- Initial contact points

Output:
- Physically valid motion plan (contacts and forces)



Pick and Rotate



Vertical Flip

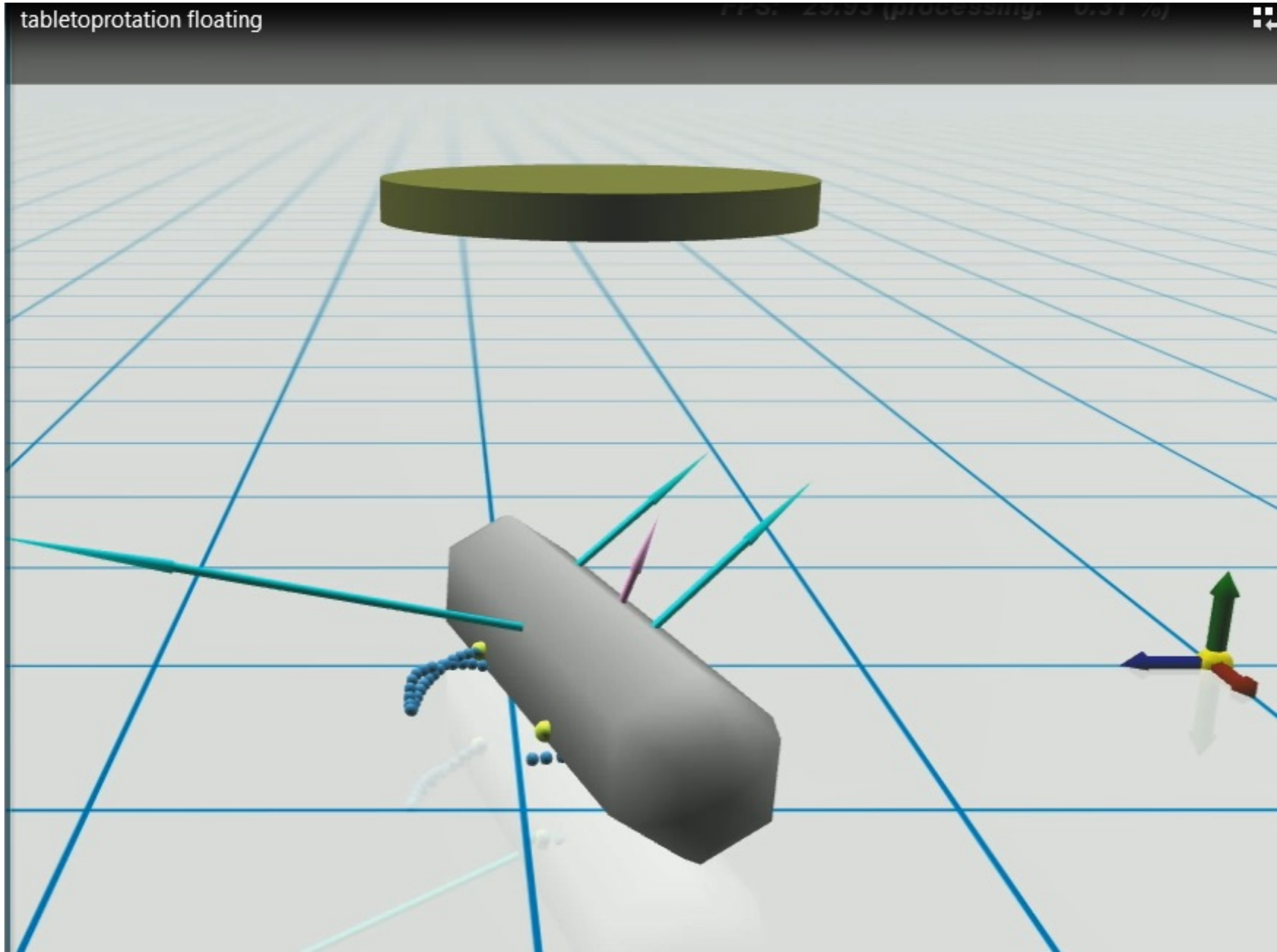
Step 1: Floating Optimization Problem

$$\mathbf{S} = \underset{\mathbf{S}}{\text{argmin}}\ \Sigma_t \Sigma_i\ w_i * L_i(t)$$

$$\text{s.t. } c_j \in [0,1]\ for\ 0 \leq t \leq T$$

$$\mathbf{S_t} = [\mathbf{x_O}\ \mathbf{f_j}\ \mathbf{r_j}\ c_j]$$

- x_O = object position + orientation
- f_j = contact force (contact j)
- r_j = contact position (contact j)
- c_j = contact invariant term



Step 1: Floating Optimization Objective Terms

- Task----specify goal of the manipulation
- Physics—force and torque balancing + friction cone constraints
- Contact Invariant terms—projection of contacts onto object surface
- Additional Regularization Terms—smooth out the motion

Task Objective Terms

$$L_{task} = \frac{1}{k}\sum_{k} ||p(k) - p_{goal}(k)||^2 + quatdist(o(k), o_{goal}(k))^2$$

- Main objective type: object pose
- Quatdist: angular distance between 2 orientations

Alternative/additional objectives:

- End effector tracking between object and target points
- Additional perturbing forces

$$L_{eeTracking}(t) = \sum_i ||p_{ee}(t) - p_{target}(t)||^2$$

Physics Terms

$$L_{physics}(t) = L_{linMomentum}(t) + L_{angMomentum}(t)$$

$$L_{linMomentum}(t) = \underbrace{\sum_i c_i(t) f_i}_{\text{Applied Force}} - \underbrace{m\ddot{x}}_{\text{Derivative of linear momentum}}$$

$$L_{angMomentum}(t) = ||\underbrace{\Sigma_i\, c_i(t) * (r_i \times f_{i,local})}_{\text{Applied Torque}} - \underbrace{(\omega \times (I^{local}_{object}\omega) + I\dot{\omega})}_{\text{Derivative of angular momentum}}||^2$$

- x = object position
- f_j = contact force (contact j)
- r_j = contact position (contact j)
- c_j = contact invariant term

Force and Contact Related Terms

Force Related Costs

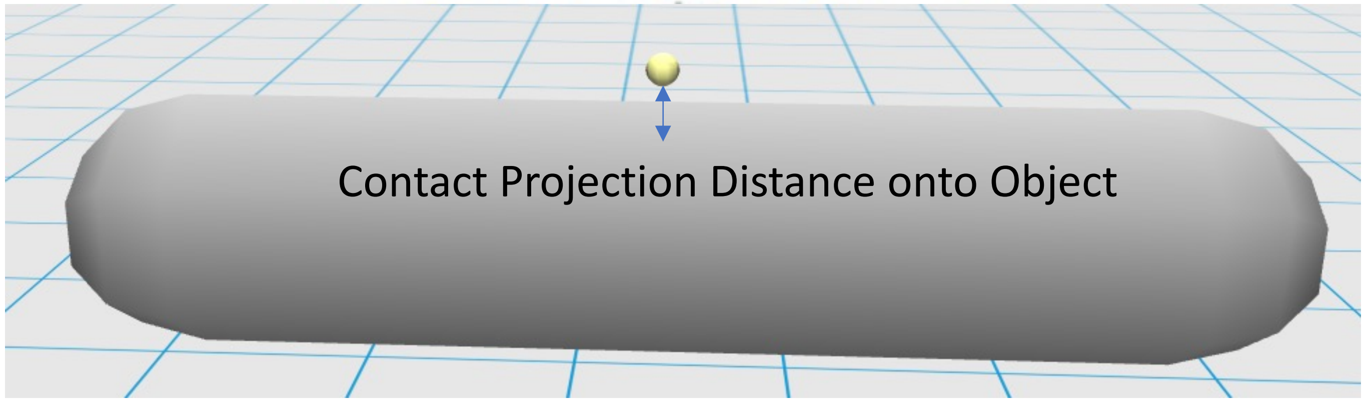
$$L_{forceReg}(t) = \sum_i ||c_i(t) f_i||^2$$

$$L_{frictionCone}(t) = \sum c_i * \exp(\alpha(||f_{i,local} - n * (f_i \cdot n_i)|| - \mu f_i \cdot n_i))$$

$f_{tangent}$ f_{normal}

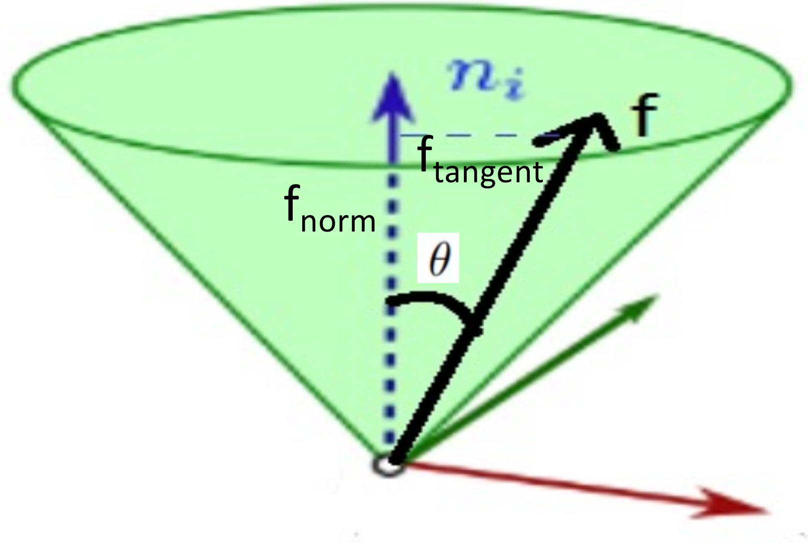
Contact Invariant Related Costs

$$L_{ci\_object}(t) = \sum c_i ||r_{proj} - r_i||^2$$



For contact i:
- f_i = contact force
- r_i = contact position (object local frame)
- n_i = object surface normal (local frame)

Alpha is a constant (sharpening factor)



$$f_{tangent} \le \mu f_{normal} \text{ i.e.}$$

$$\theta = \tan^{-1}\frac{f_{tangent}}{f_{normal}} \le \tan^{-1}\mu$$

Additional Regularization Terms

$$L_{floatingContactAccel}(t) = \sum_i ||((r_i(t + t_{phys}) - 2 * r_i(t) + r_i(t - t_{phys}))/(t_{phys} * t_{phys})||^2$$

Acceleration of contact: finite differences

$$L_{accelerationRegularization}(t) = \sum_i \ddot{x}^2$$

Object acceleration: finite differences

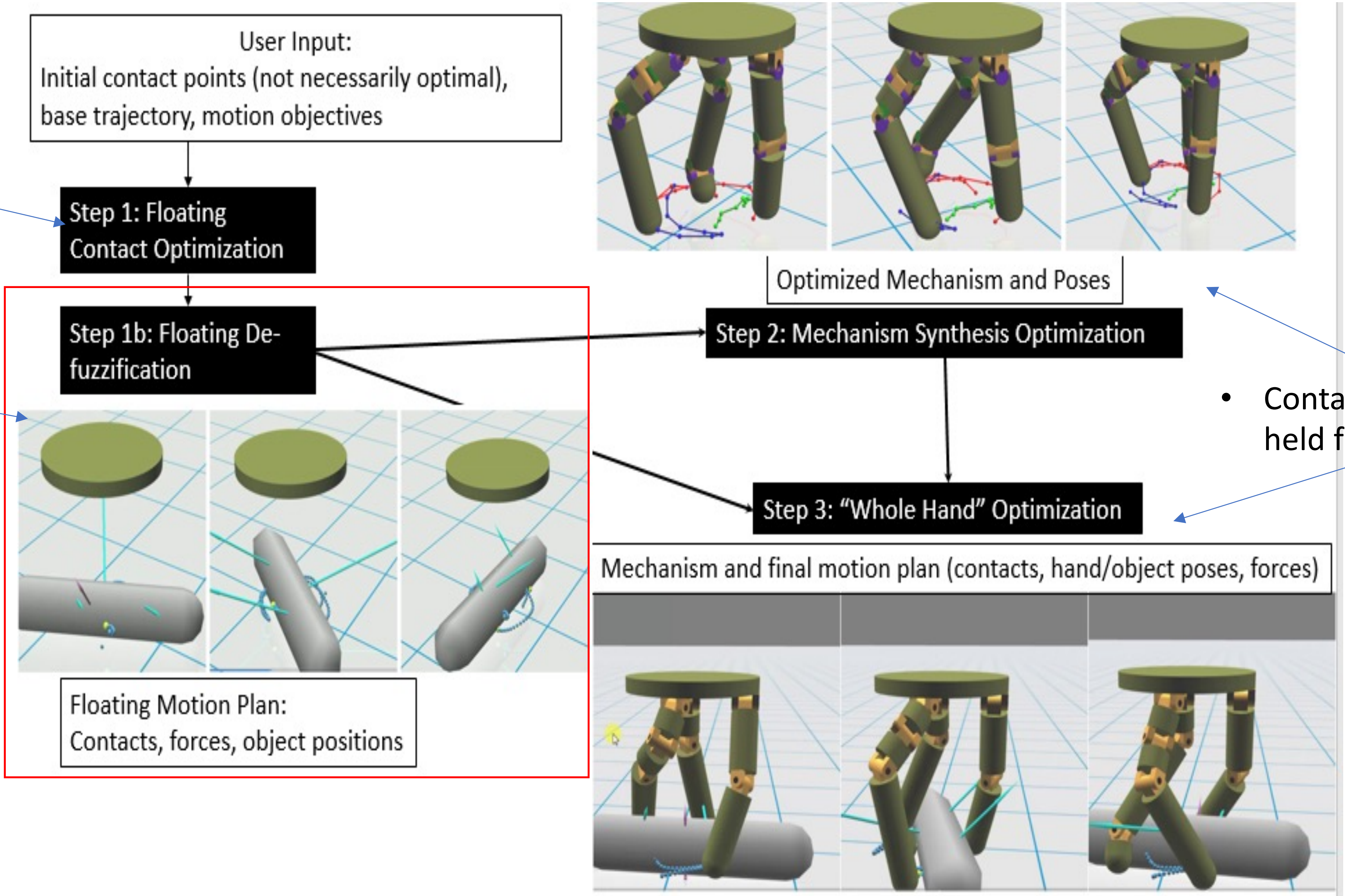
$$L_{angularAccelerationRegularization}(t) = \sum_i (\omega \times (I_{world}\omega) + I_{world}\dot{\omega}/t_{phys})^2$$

Angular Momentum derivative

Floating Post-Processing

- Continuous Contact Variables

- Binarize contact variable
- Threshold and reoptimize



- Contact variables held fixed (binary)

Step 2: Mechanism Synthesis

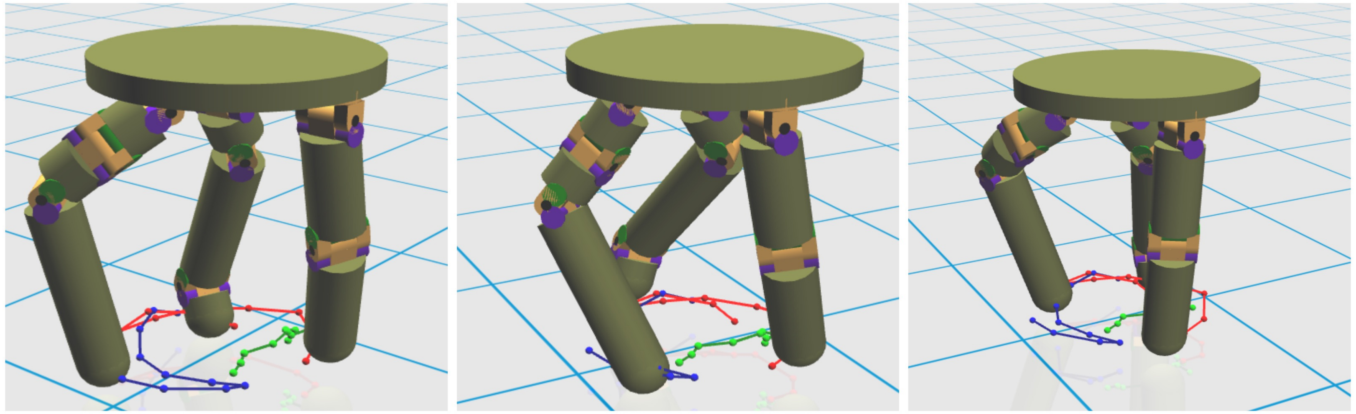
Step 2: Mechanism Synthesis

Floating Optimization

Contact Motion Plan

Synthesis Optimization:
Joints per finger, joint axes, segment lengths, finger positions on base, hand poses

-Fingers track individual contact trajectories
-Independently controlled joints



Output: Optimized Mechanism + Poses

Continuous Synthesis Optimization

$$\mathbf{M} = \underset{\mathbf{M},\mathbf{Q},\mathbf{P}}{\operatorname{argmin}} \ \Sigma_k \Sigma_i \ w_i * L_i(k)$$
$$\text{for } k \in \{1, 2, ..., N_{keyframes}\}$$

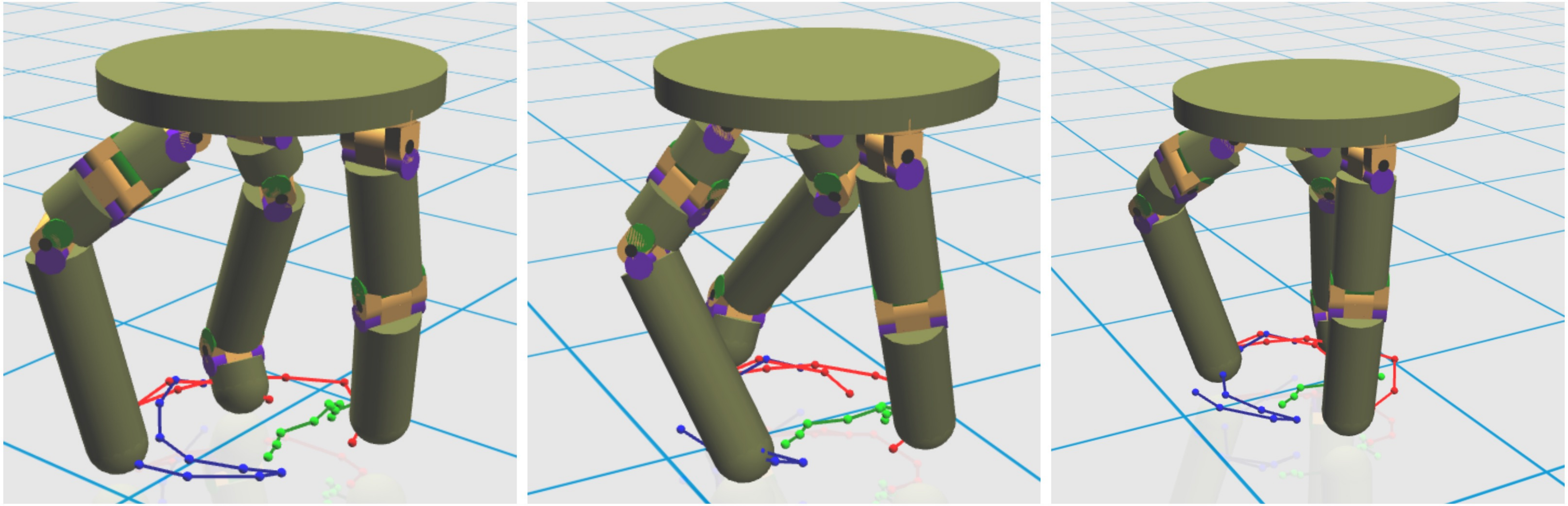
- Morphological parameters M:
  - -finger lengths
  - -joint axes
  - -locations of fingers on the base
- Joint positions Q (hand poses at each keyframe)
- Contact points P (on fingertips)

Synthesis Objective Terms

Contact Point Costs

$$L_{eeTarget}(k) = \sum_i c_i * ||p_i - p_{target}||^2$$

$$L_{fingerContactDistSurface}(k) = \sum ||p_{proj} - p_i||^2$$

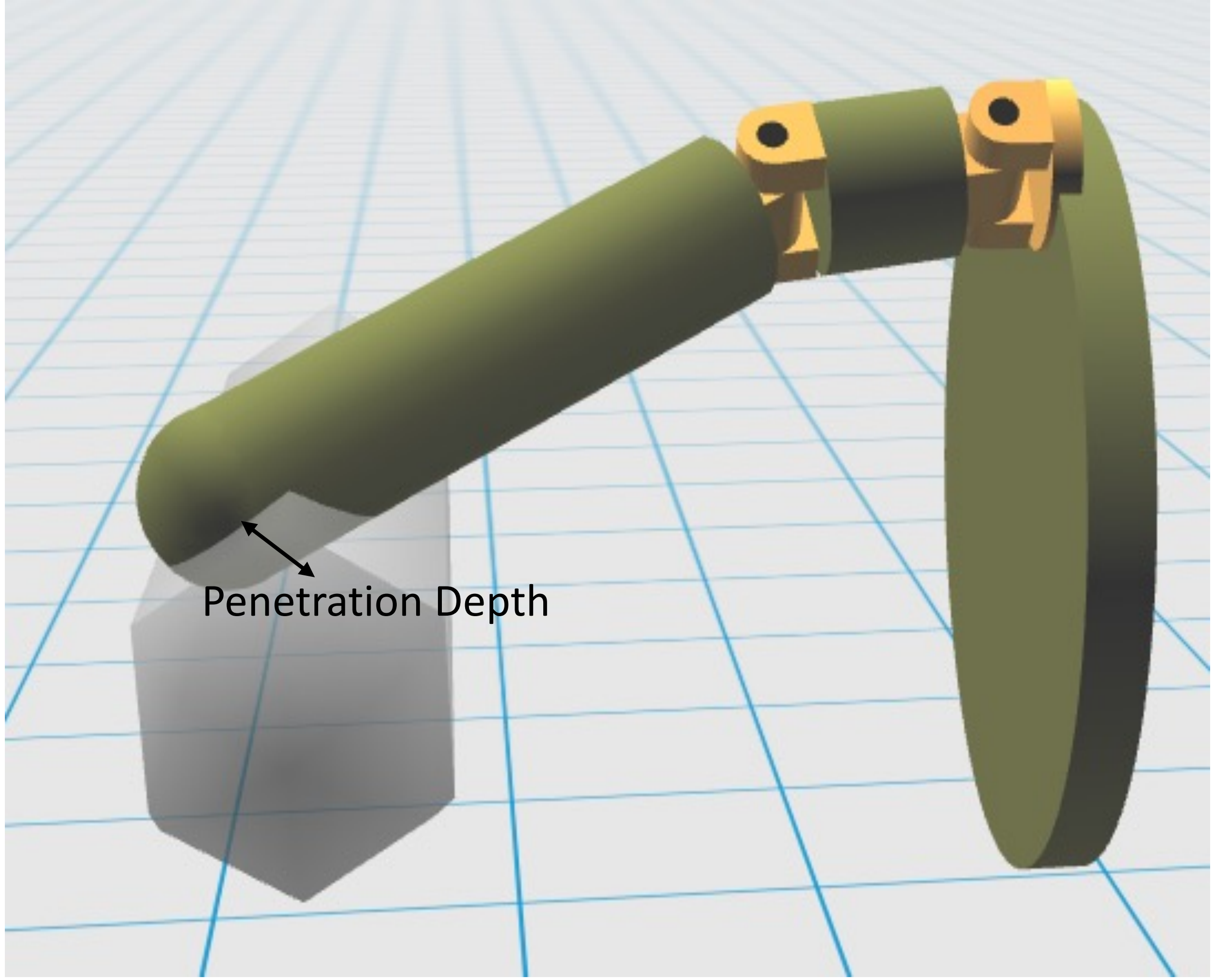


Synthesis Objective Terms

Collision Penalties

$$g(x) = \begin{cases} 0 & x \leq -\epsilon \\ \frac{x^3}{6\epsilon} + \frac{x^2}{2} + \frac{\epsilon x}{2} + \frac{\epsilon^2}{6} & -\epsilon \leq x \leq \epsilon \\ x^2 + \frac{\epsilon^2}{3} & \epsilon \leq x \end{cases}$$

$$L_{collision}(k) = \sum_{i,j \in bodies} g(penetration(body_i, body_j))$$

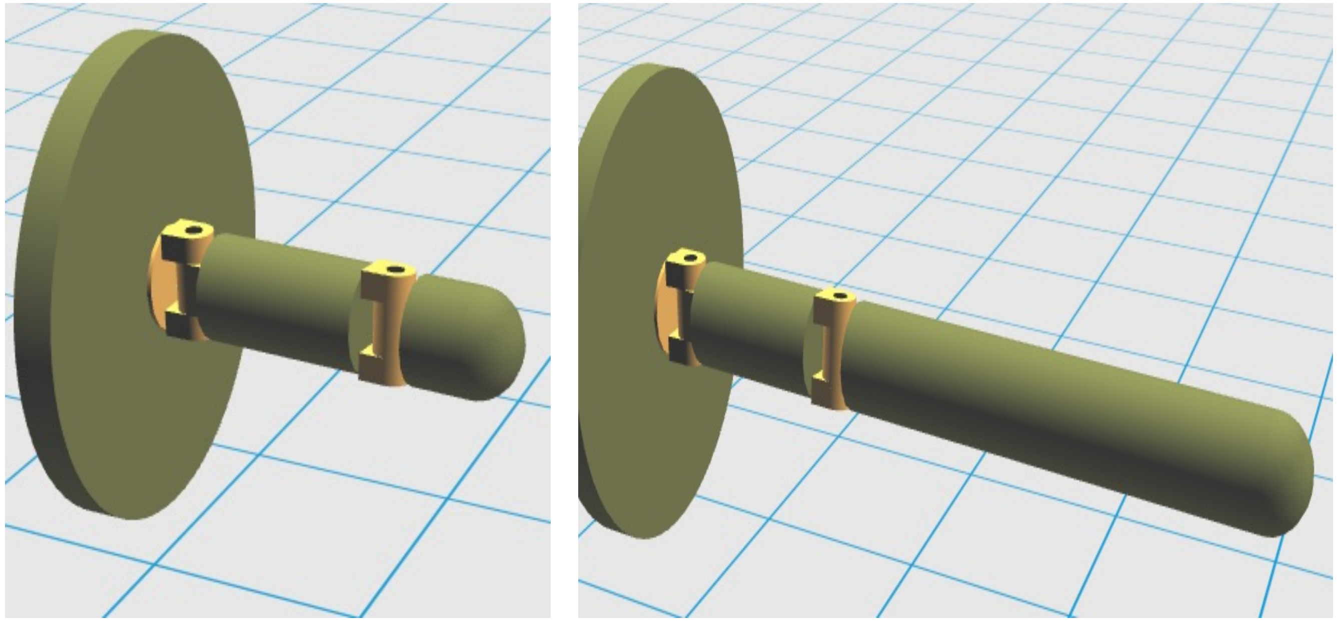


Additional Costs

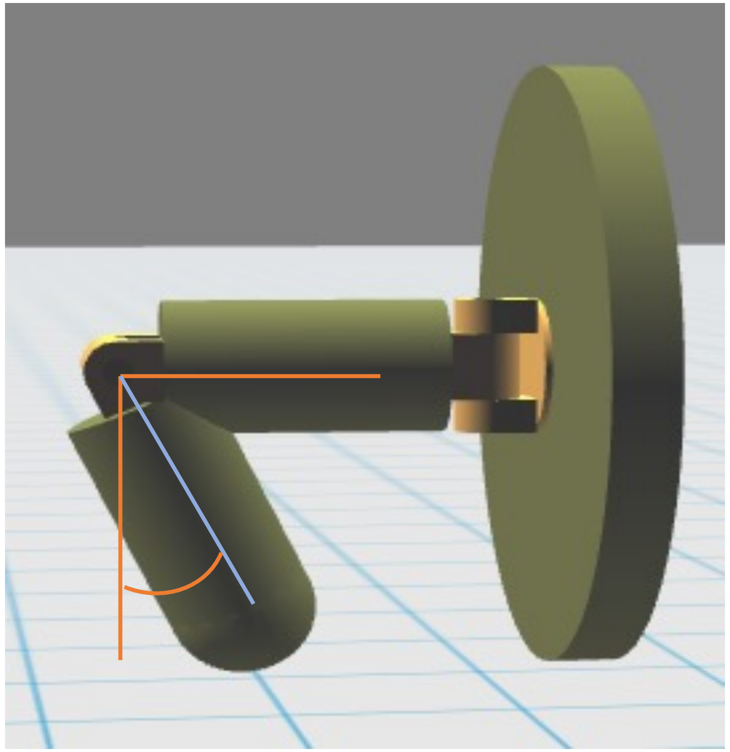
$$L_{fingerLengthRegularization} == \sum_i (l_i)^2$$

$$L_{jointLimits}(k) = \sum_{i \in keyframes} \sum_{a \in joints} g(a(i) - a_{max}) + g(a_{min} - a(i))$$

$$L_{fingerMinLengthCost} = \sum_i g(l_{min} - l_i)$$



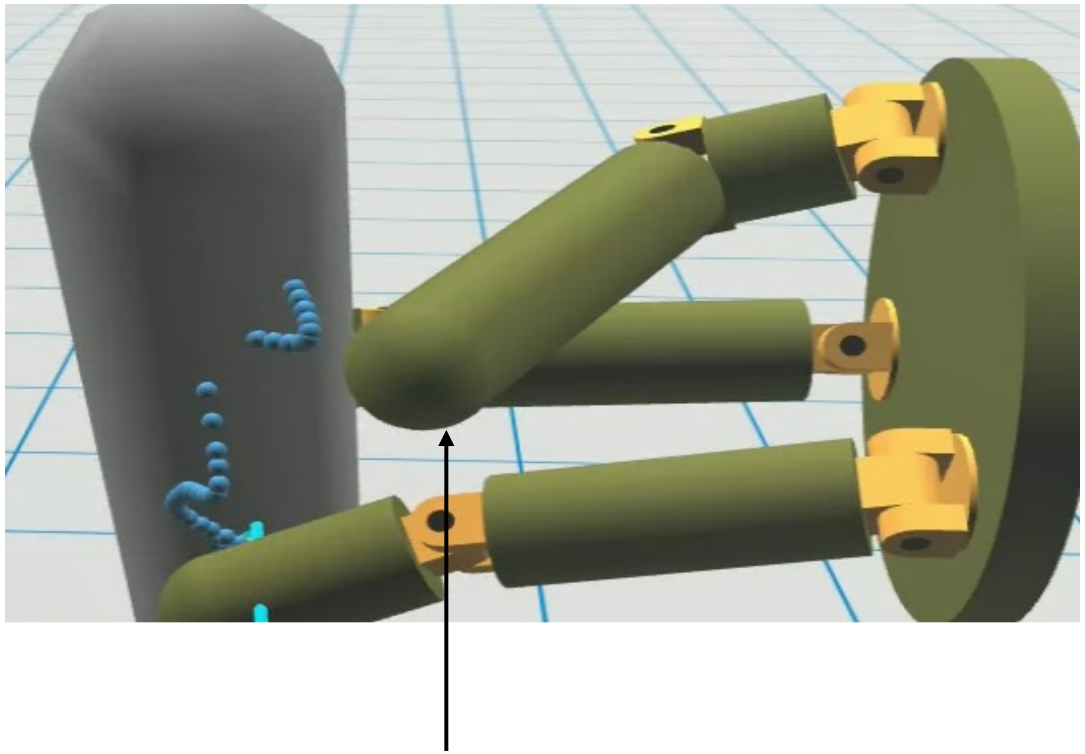
Distal link: min length and a large length



Joint Limit Violation

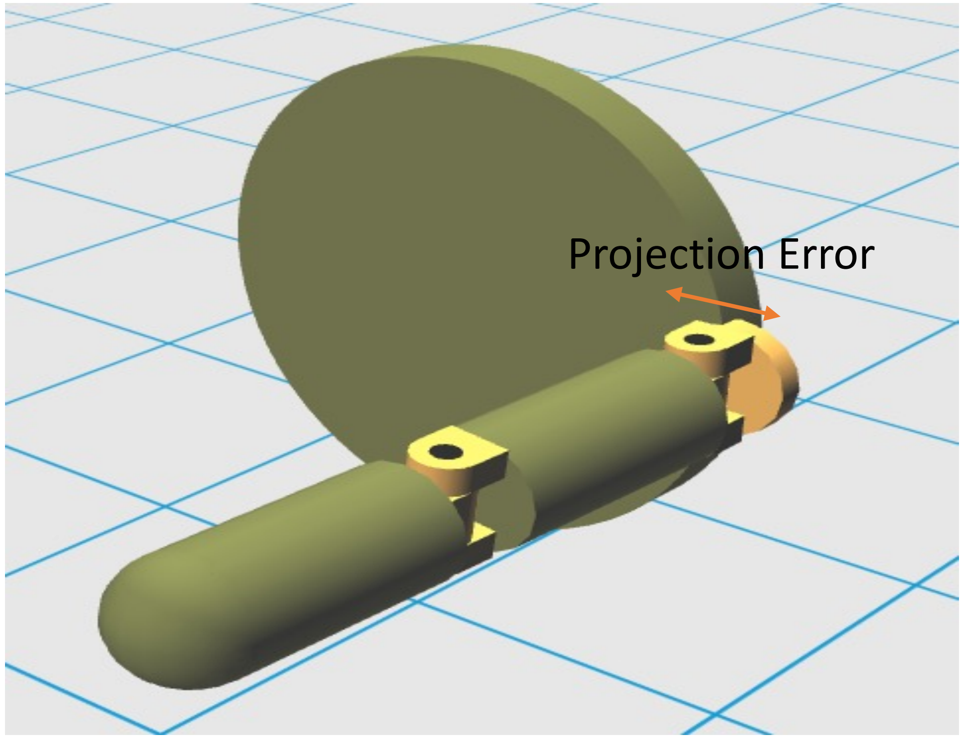
Additional Costs

$$L_{fingerAcceleration}(k) == \sum_i (1 - c_i) * \ddot{x}_i{}^2$$



Lifted finger transitions smoothly from one side to the other

$$L_{fingerPositions} = \sum_i ||proj_{base}(b_i) - b_i||)^2$$

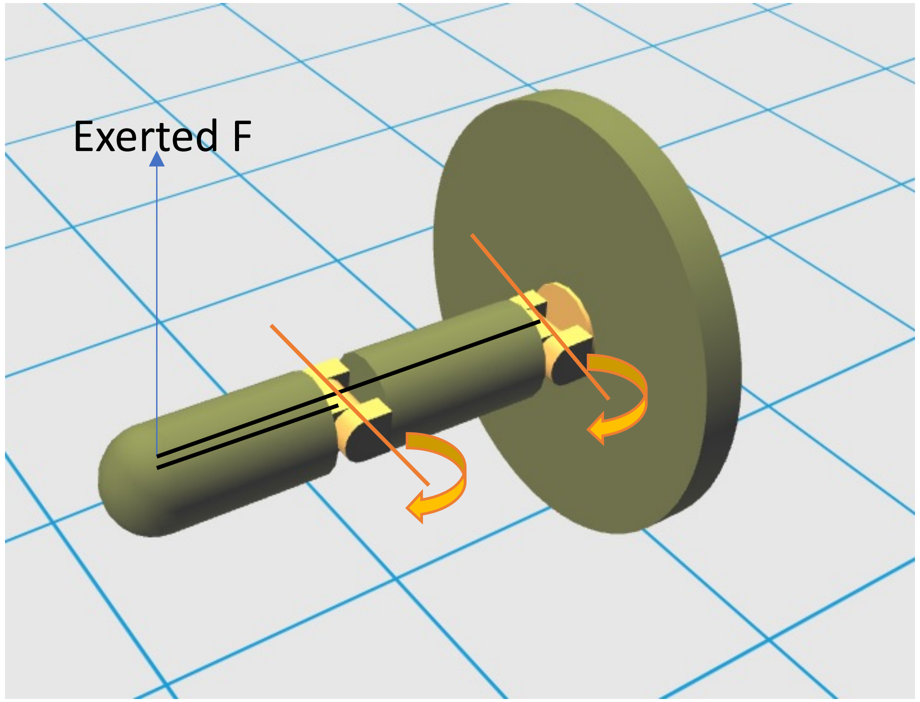


Controllability Constraints

Jacobian Null Space:

Let E = $\{e_0,\ldots,e_k\}$ be an orthonormal basis of the Jacobian null space:

$$L_{jacNull} = \sum_i c_i * \sqrt{\sum_k (f \cdot e_k)^2}$$



Torque Regularization:

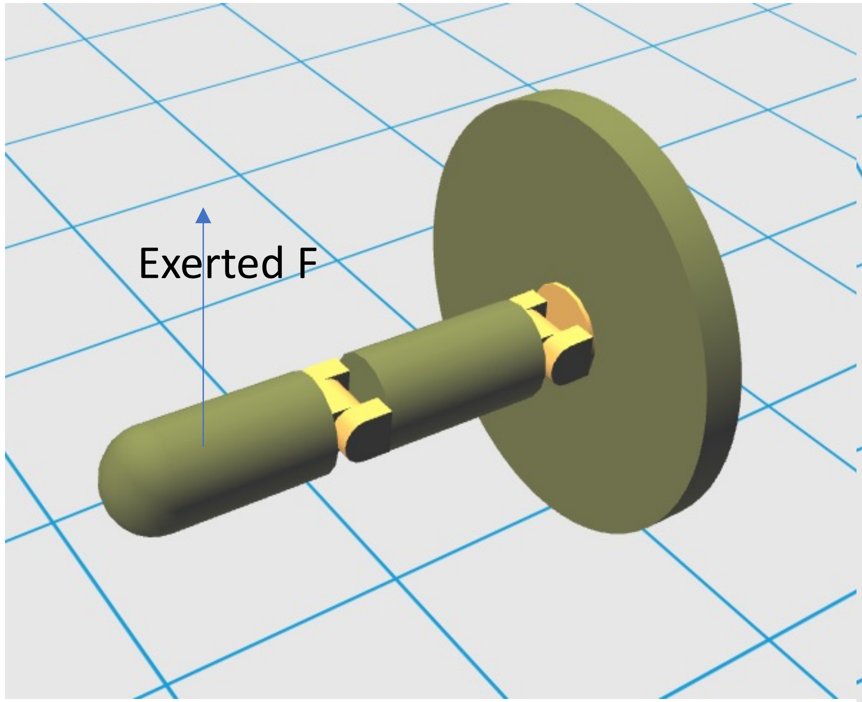
$$F = T \times r_{perp}/\|\vec{r_{perp}}\|^2 + k * r_{perp}$$

$$F = X * \alpha \longrightarrow \vec{\alpha} = (X^T X + \lambda^2 I)^{-1} X_T F$$

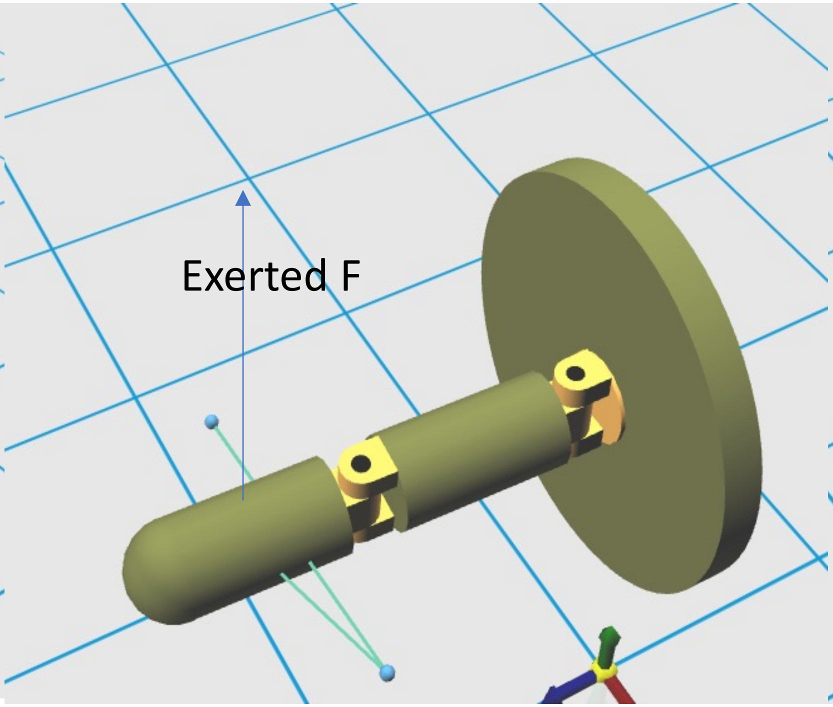
where X is the matrix consisting of column vectors $T \times r_{perp}/\|\vec{r_{perp}}\|^2$

$$L_{torque} = \|\vec{\alpha}\|^2$$

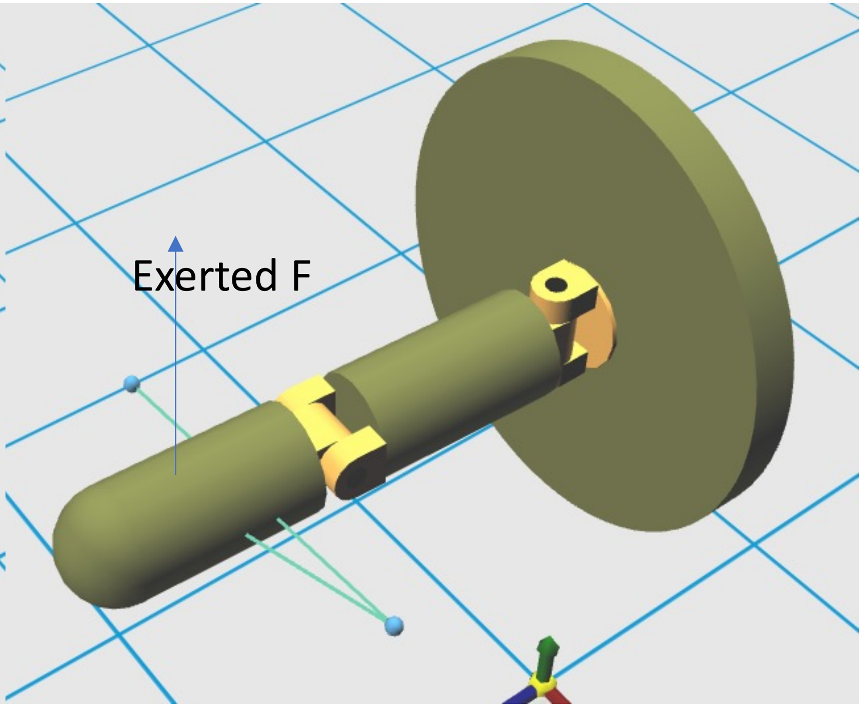
Controllability Constraints Demonstration



1. Finger held still: optimal joints

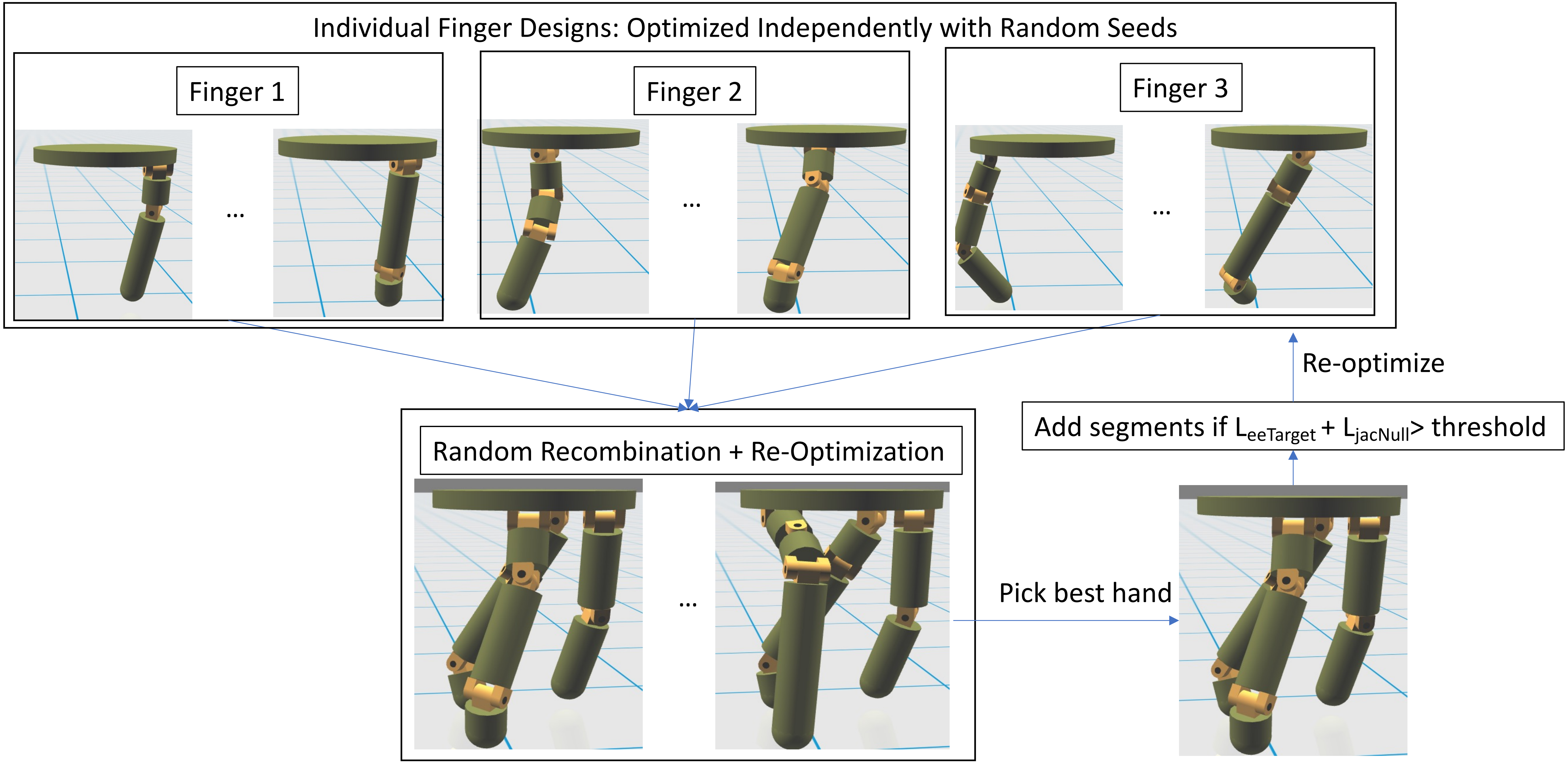


2. Finger rotates in plane
Joints slightly off axis:
$L_{jacNull} = 0$
L_{torque} very high



3. Finger rotates in plane:
Optimal joint configuration

Synthesis Design Loop



Step 3: Whole Hand Optimization

Whole Hand Optimization Problem

We wish to find a trajectory $\mathbf{S} = \{\mathbf{S_1}, \mathbf{S_2}, ..., \mathbf{S}_{N_{keyframes}}\}$ such that

$$\mathbf{S} = \underset{\mathbf{S}}{\text{argmin}}\ \Sigma_t \Sigma_i\ w_i * L_i(t)$$
$$\text{s.t. } c_j \in [0,1]\ for\ 0 \leq t \leq T$$

- Adjusts the motion so it fits to the designed hand
- Uses floating objectives + additional objectives
- Also optimize for robot poses **q**
- Morphology stays fixed

Additional Optimization Terms

Additional terms (from floating) adapted for hand:

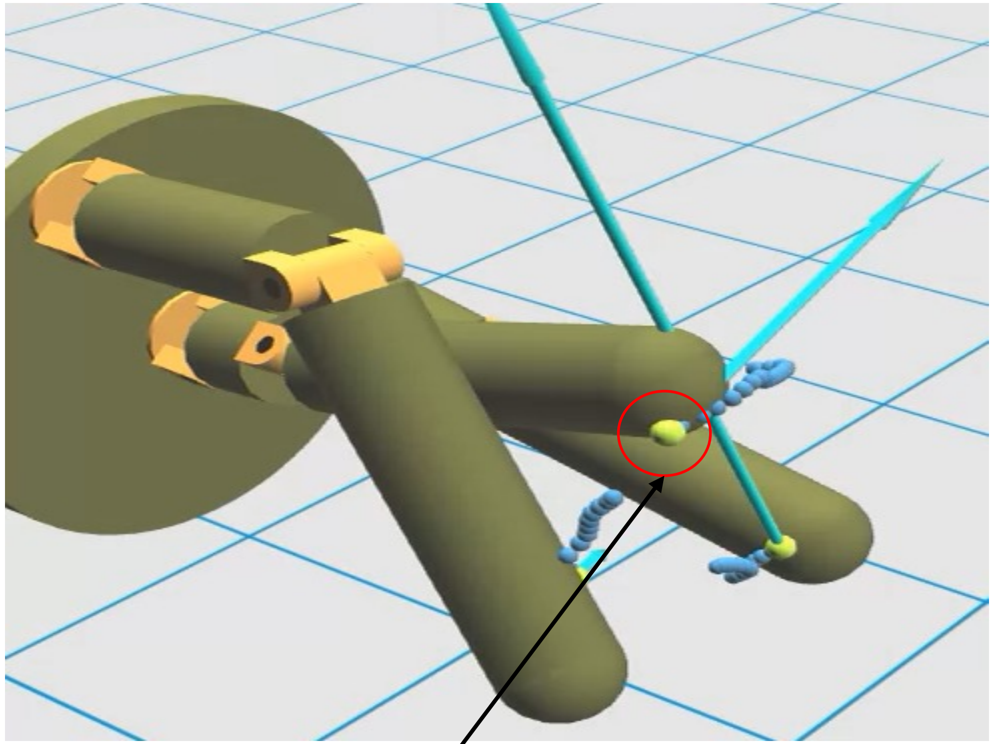
Contact projection onto fingertip surface

$$L_{ci\_finger}(t) = \sum_i c_i ||r_{proj} - r_i||^2$$

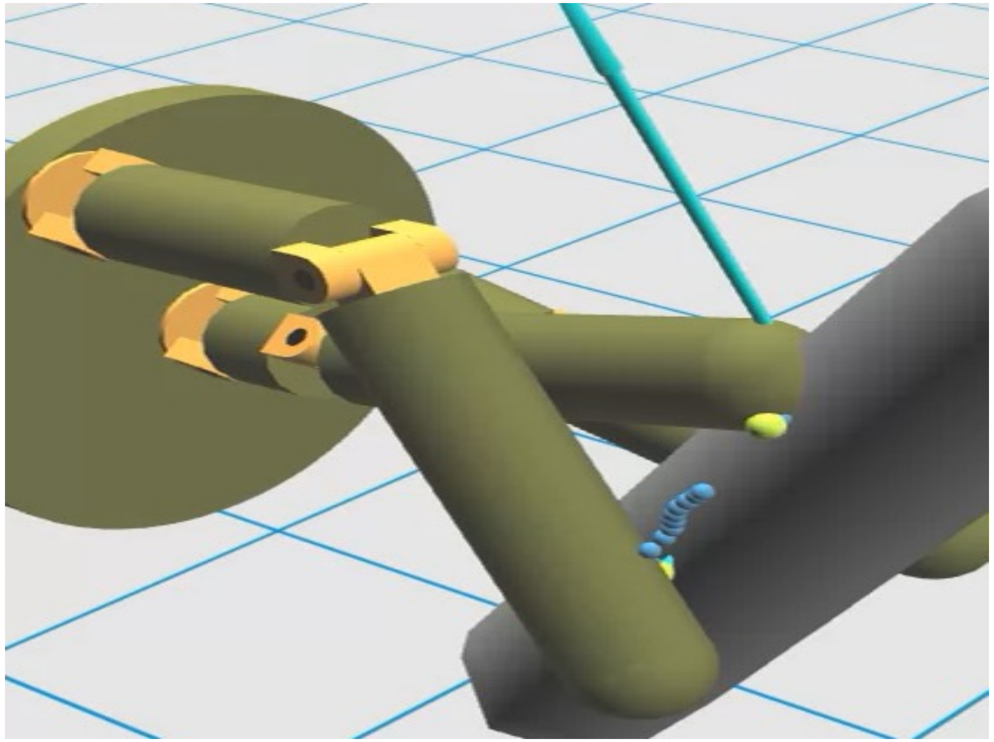
Friction Cone wrt fingertip surface

$$L_{frictionCone}(t) = \sum c_i * \exp(\alpha(||f_{i,local} - n * (f_i \cdot n_i)|| - \mu f_i \cdot n_i))$$

Hand Friction Cone Demonstration (without term)



Contact way outside friction cone w.r.t. finger



Caused by (small) errant collision with object

Additional Optimization Terms

Terms copied over from the synthesis step:

Controllability constraints

$$L_{jacNull} = \sum_i c_i * \sqrt{\sum_k (f \cdot e_k)^2}$$

$$L_{torque} = \|\vec{\alpha}\|^2$$

Collision (includes ground, hand, object, external objects)

$$L_{collision}(k) = \sum_{i,j \in bodies} g(penetration(body_i, body_j))$$

Other:

$$L_{fingerAcceleration}(k) == \sum_i (1 - c_i) * \ddot{x_i}^2$$

$$L_{jointLimits}(k) = \sum_{i \in keyframes} \sum_{a \in joints} g(a(i) - a_{max}) + g(a_{min} - a(i))$$

Slippage Terms

Slippage w.r.t. object

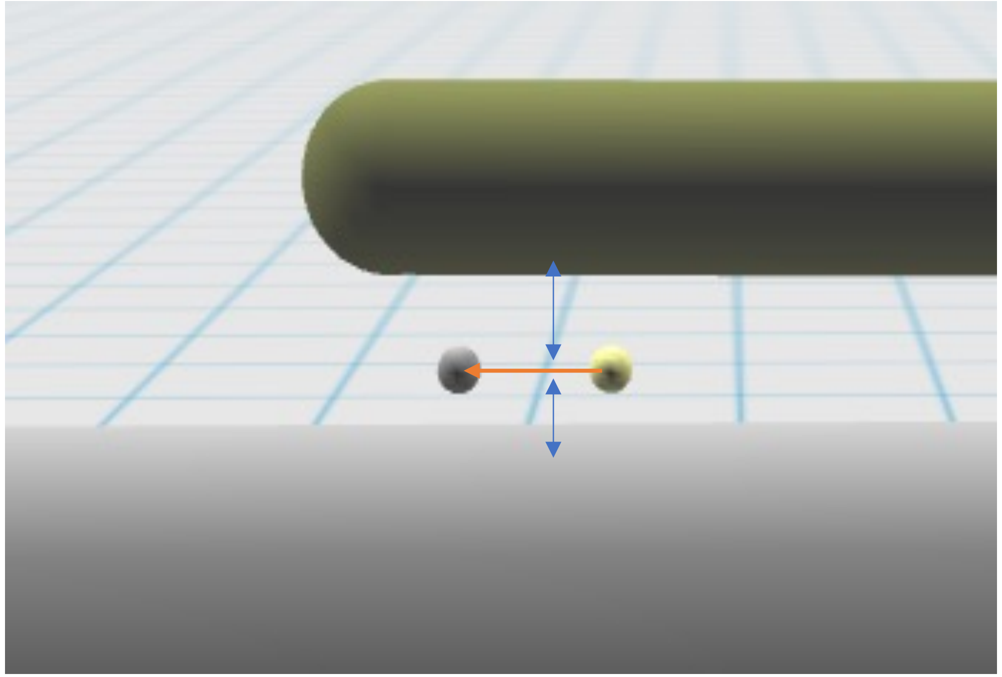
$$e_{object}(i,t) = r_{i,proj\_object}(t) - r_i(t)$$

$$L_{ci\_object\_slippage}(t) = \sum_i ||c_i f_i||^2 * ||(\dot{e}_{object}(i,t))||^2$$

Slippage w.r.t. finger

$$e_{finger}(i,t) = r_{i,proj\_finger}(t) - r_i(t)$$

$$L_{ci\_finger\_slippage}(t) = \sum_i ||c_i f_i||^2 * ||(\dot{e}_{finger}(i,t))||^2$$



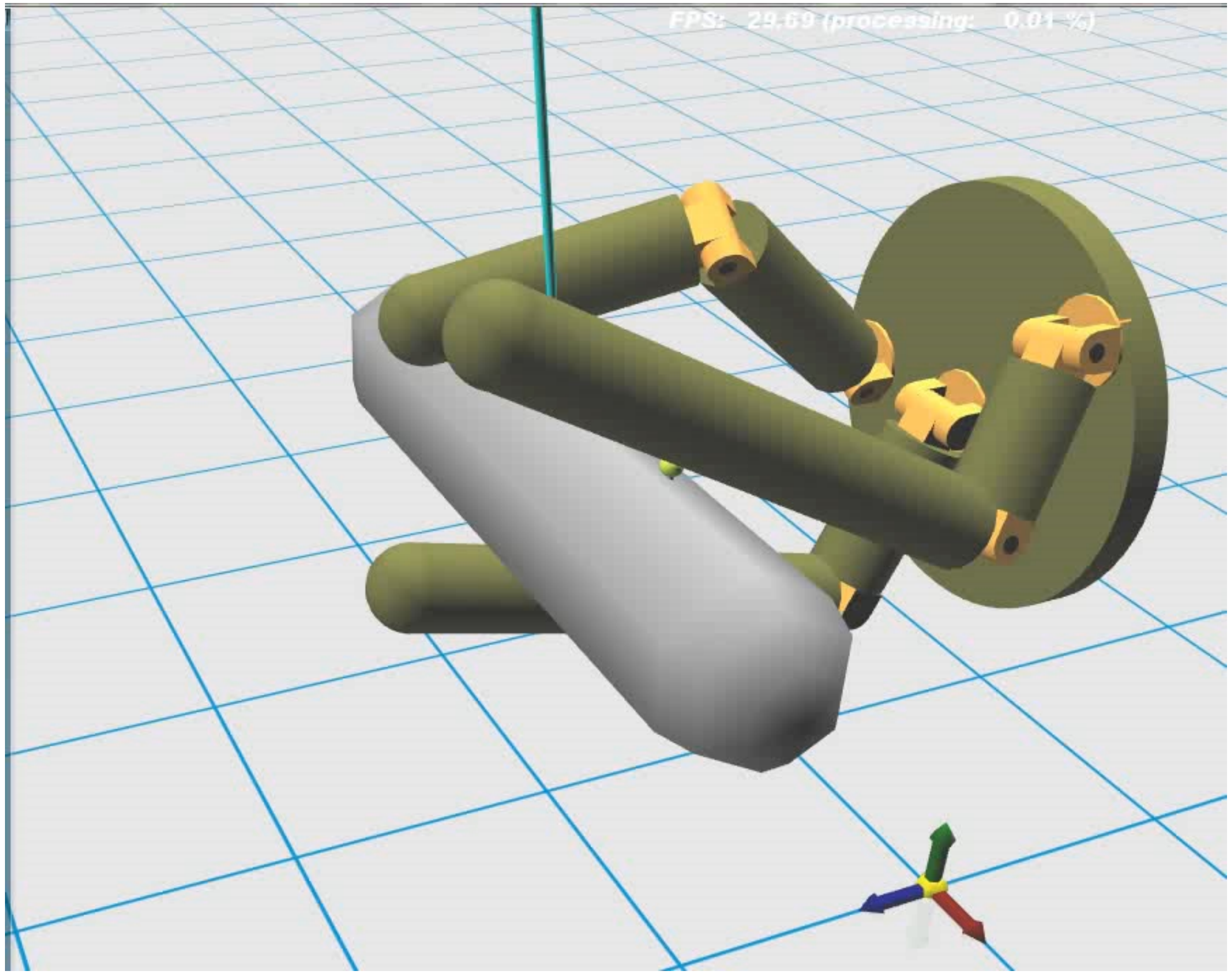
Zero slip penalty

Bottom Line: distance slipped on object = distance slipped on fingertip (w.r.t. world frame)

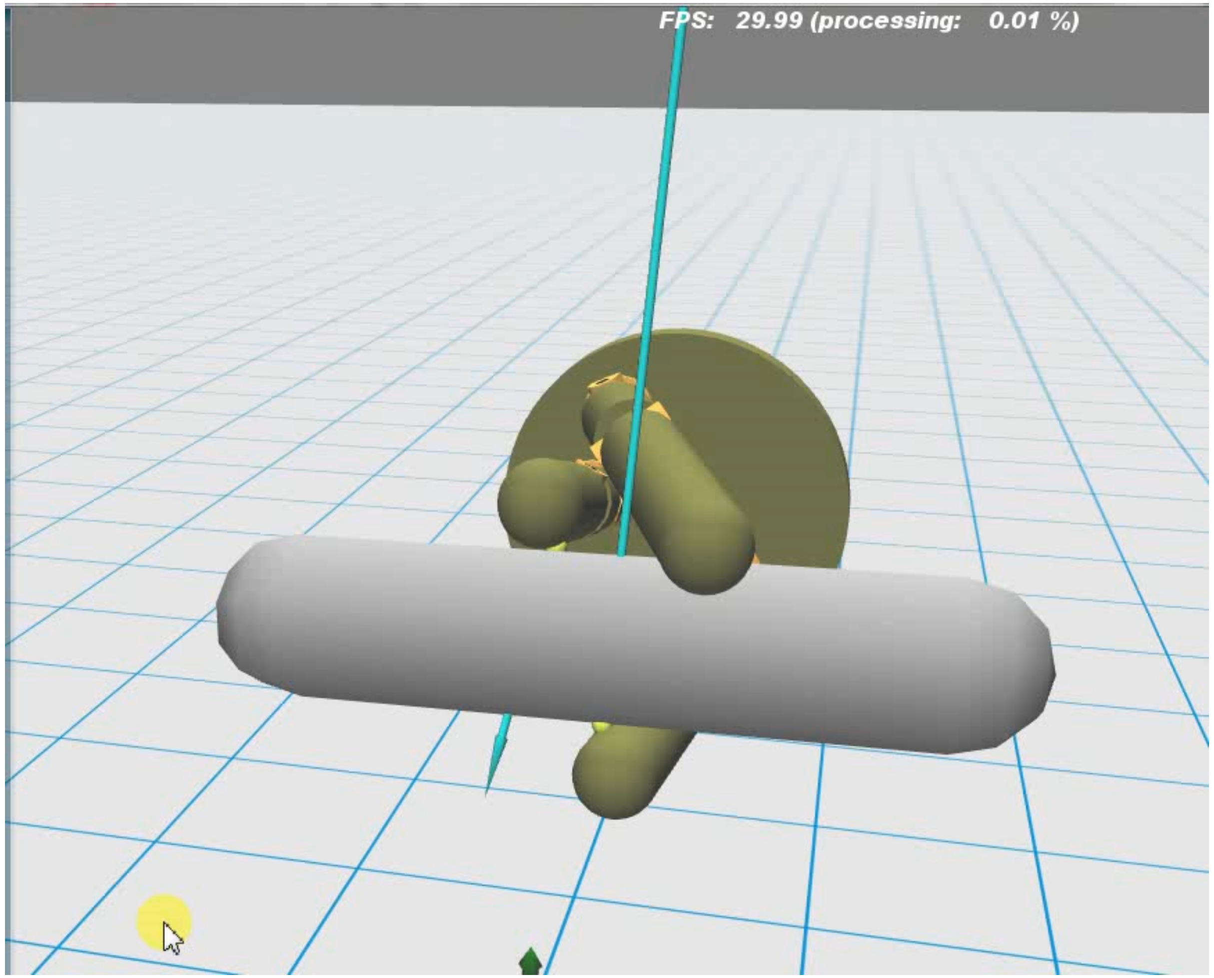
Slip directions w.r.t world frame line up

Not a complete model, but helpful

Simple Manipulations

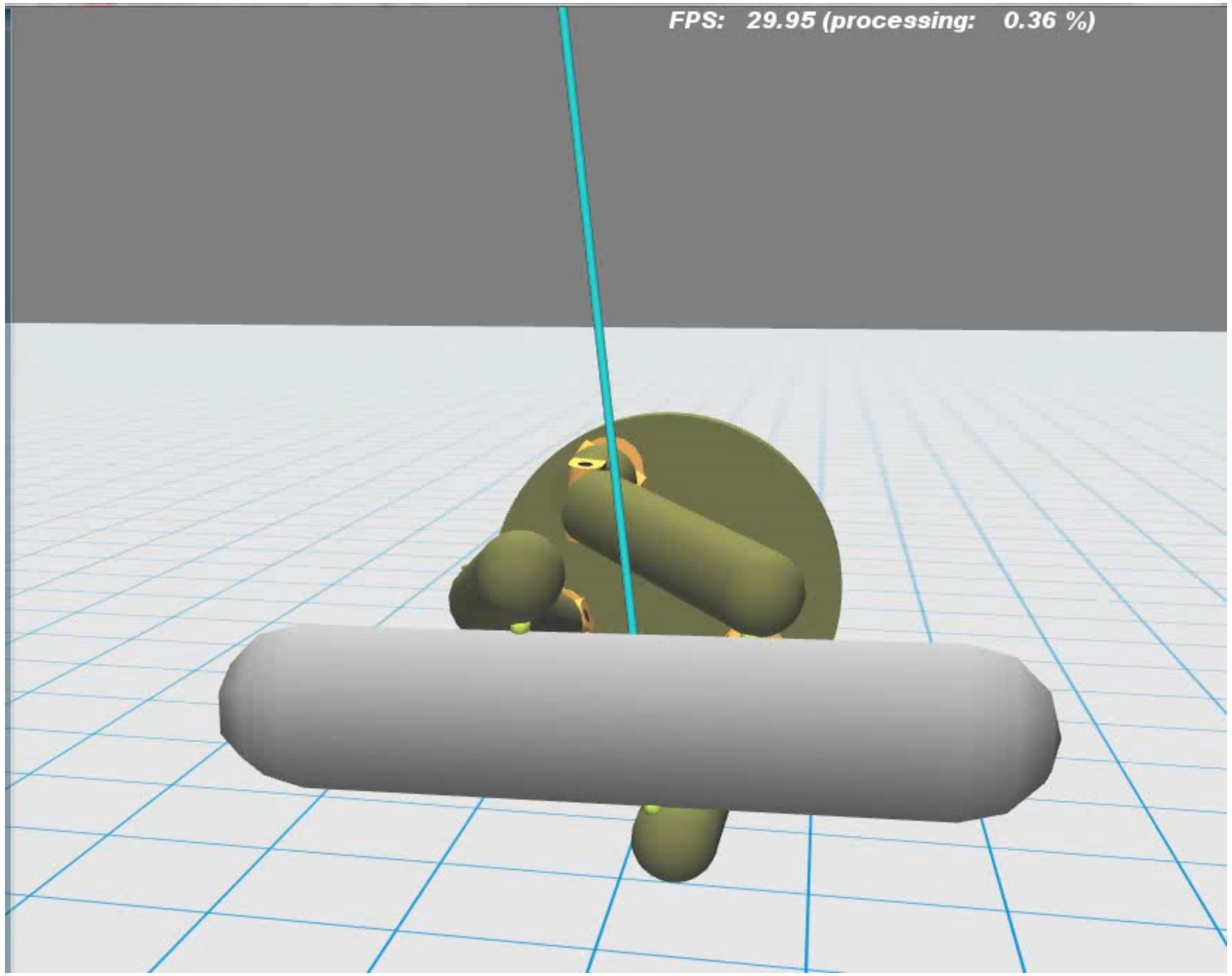


Translate

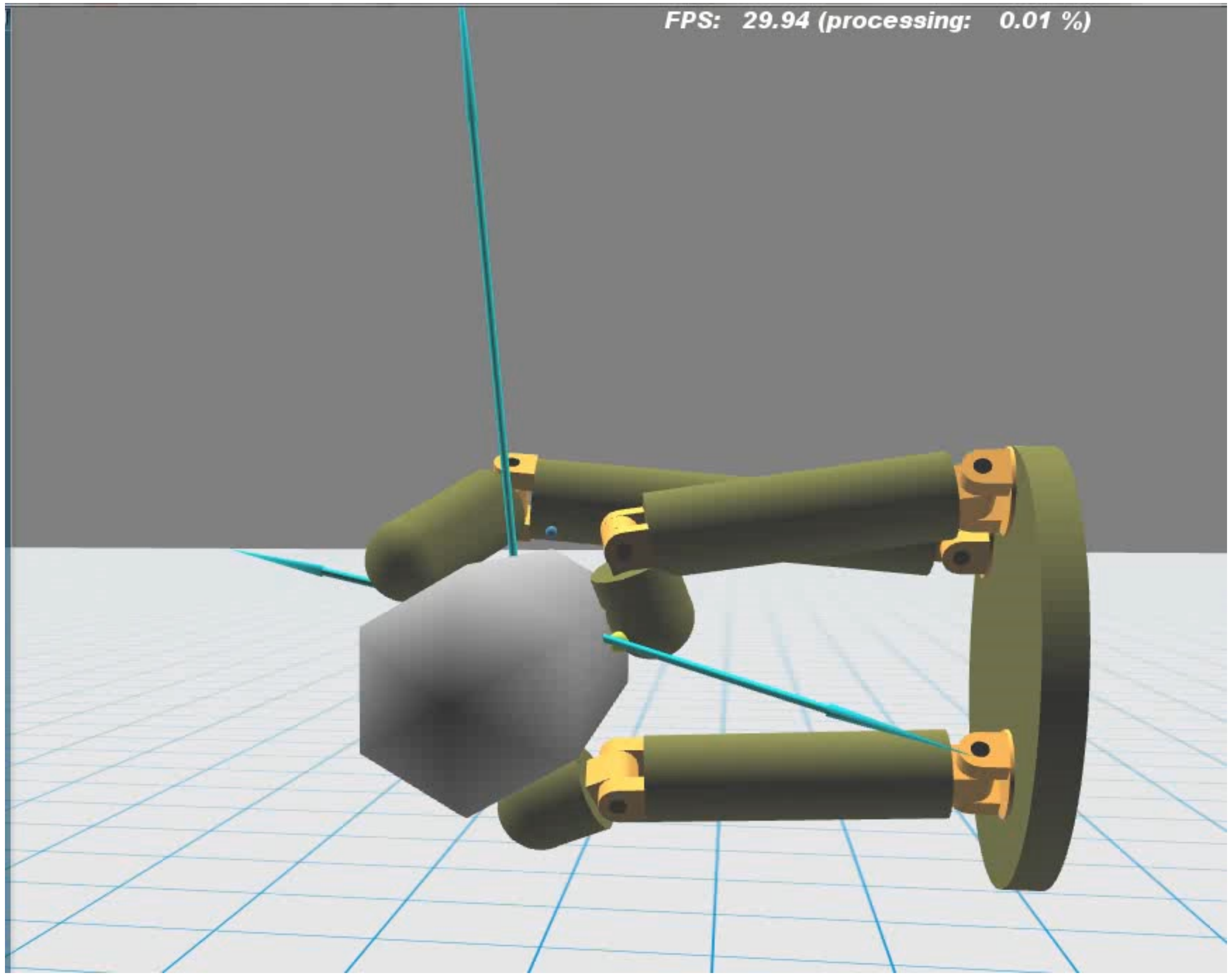


Vertical Rotate

Examples

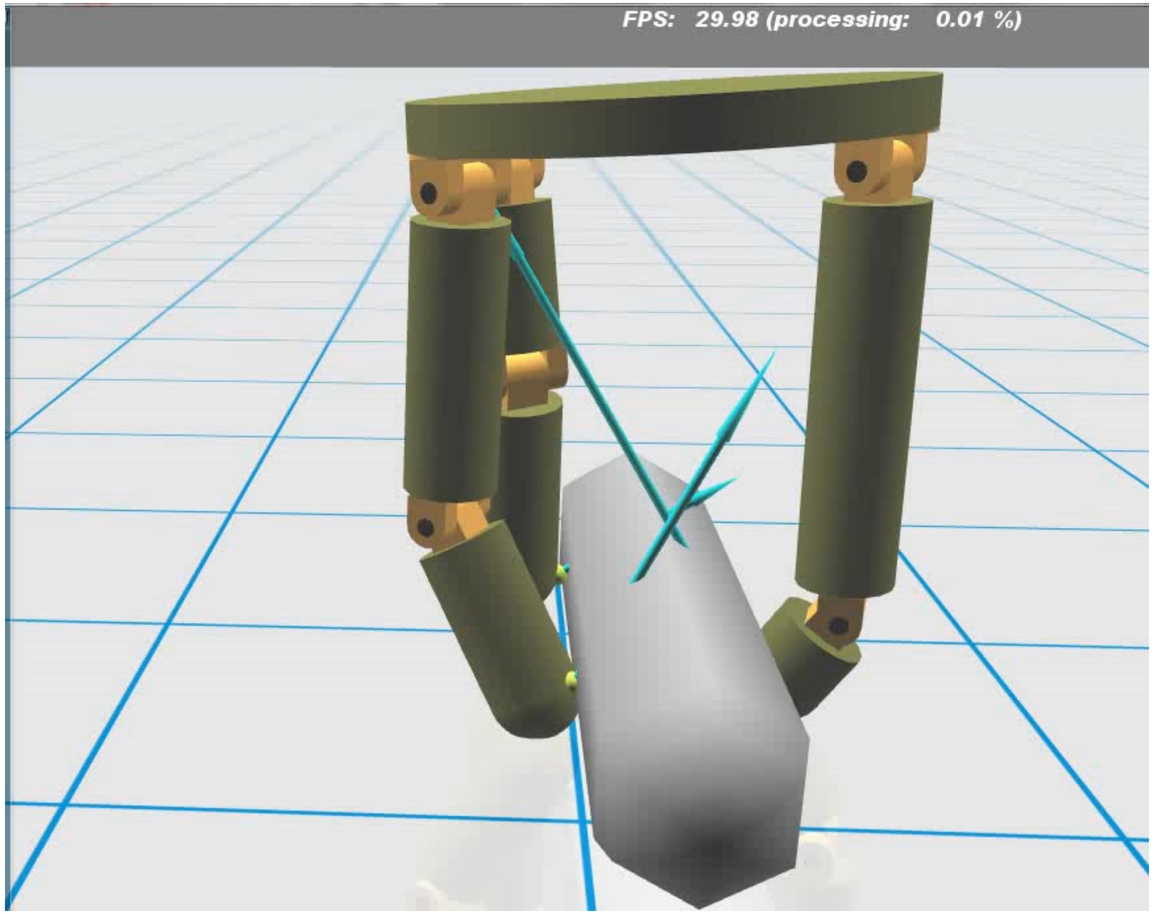


180 Rotation

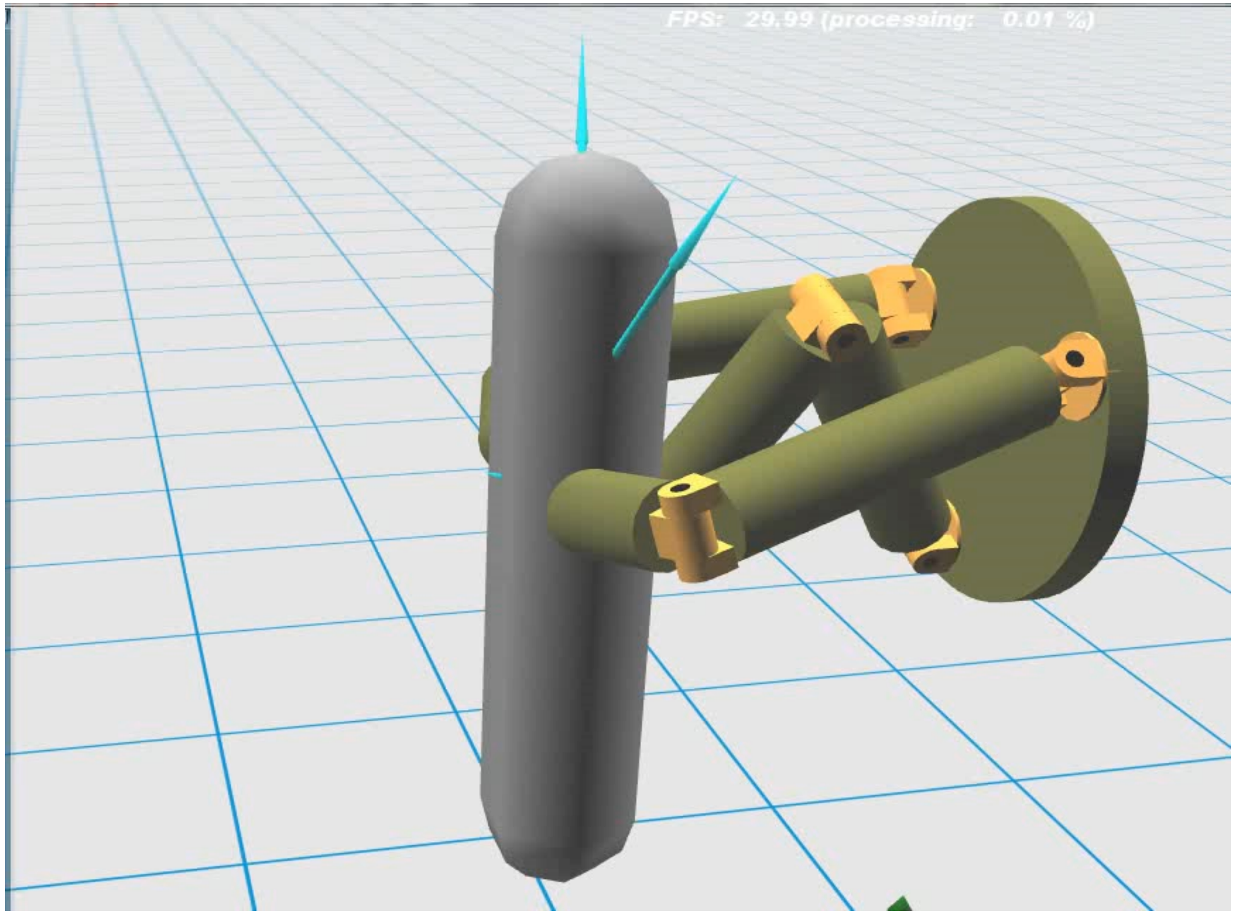


Rotate and Bow Out

More Examples

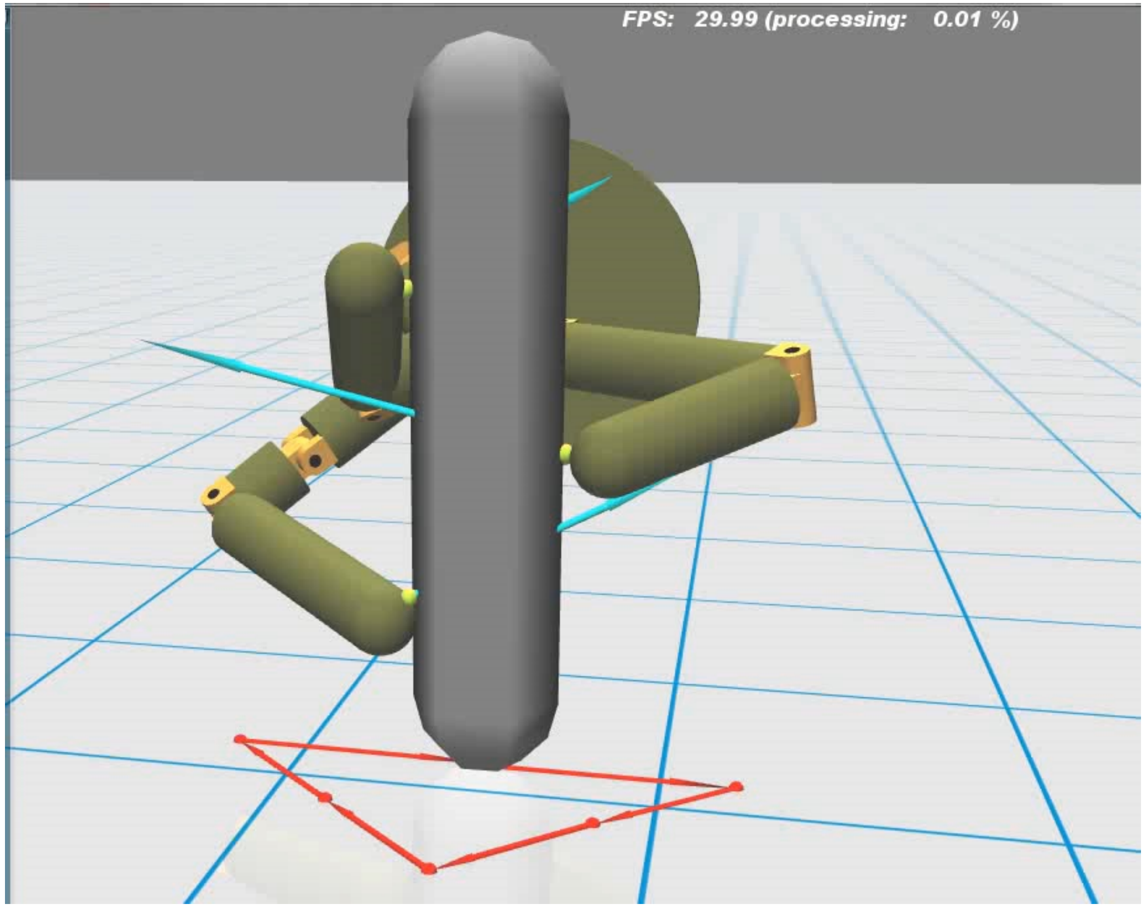


Pick up and rotate

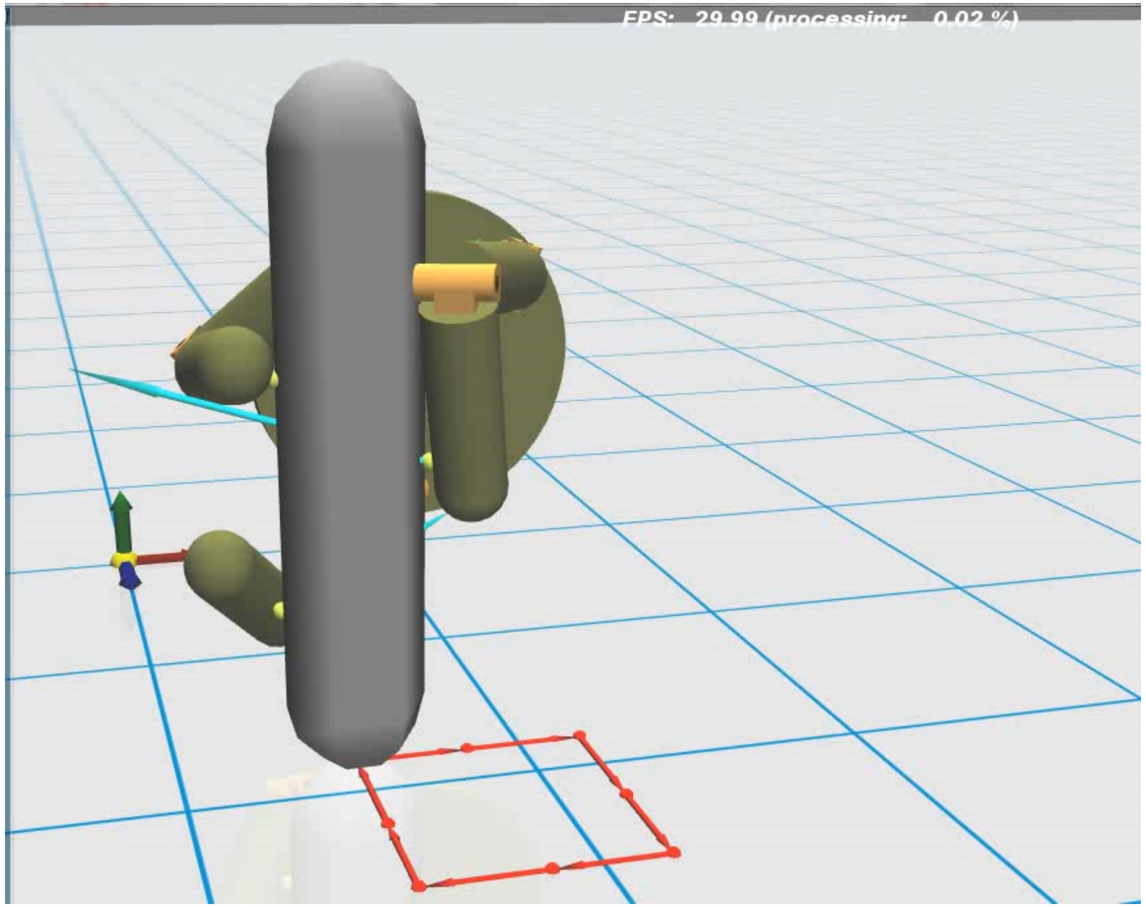


Vertical flip

Alternative Objective: Drawing



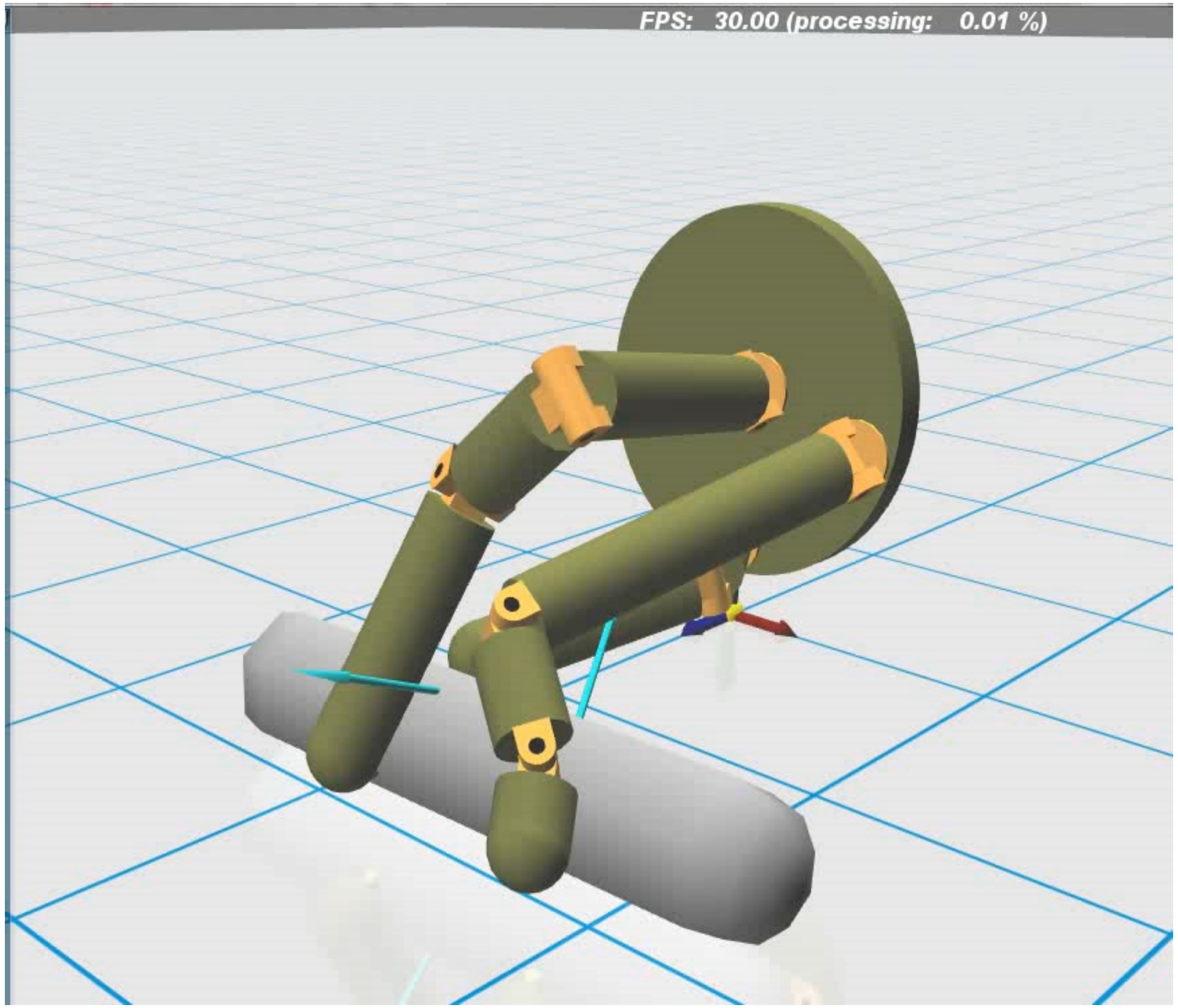
Draw triangle



Draw box

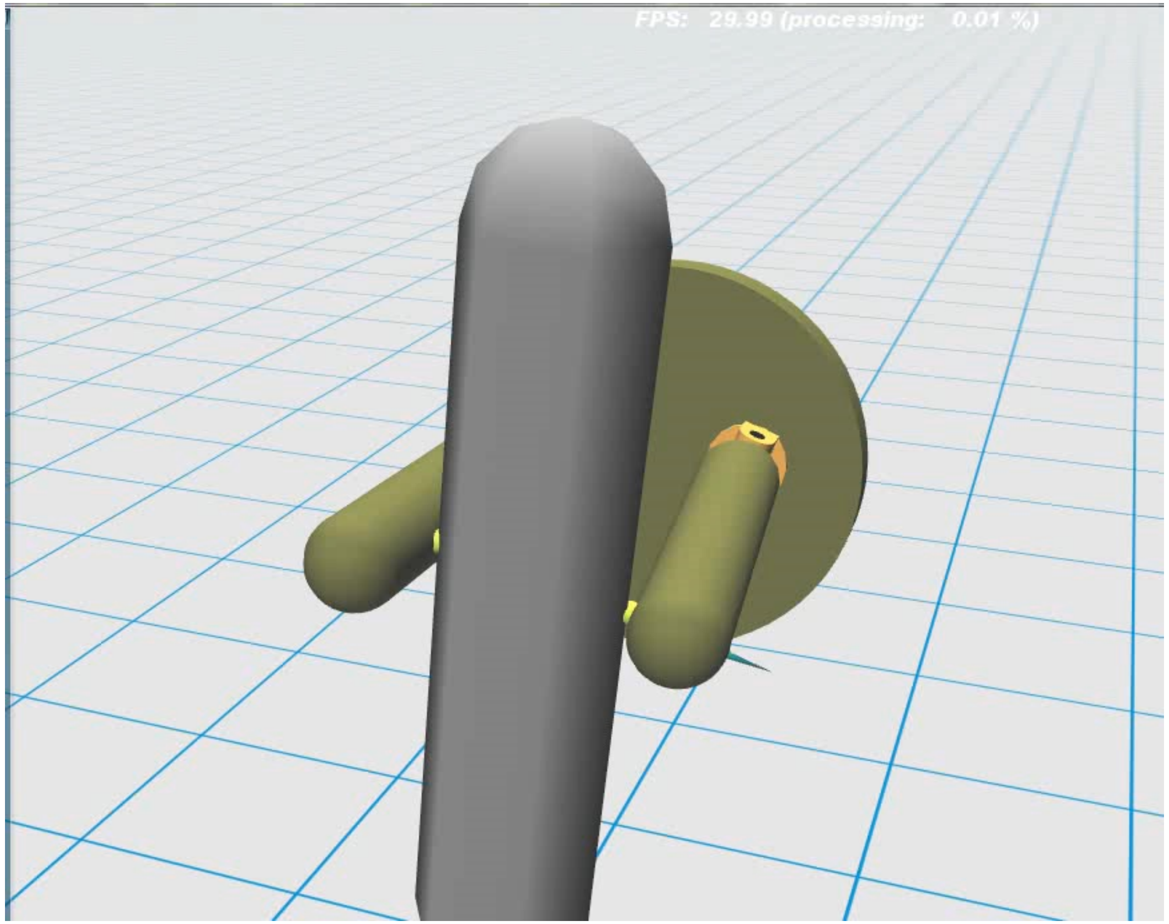
Tabletop Rotation: Two versions

Tabletop overtop

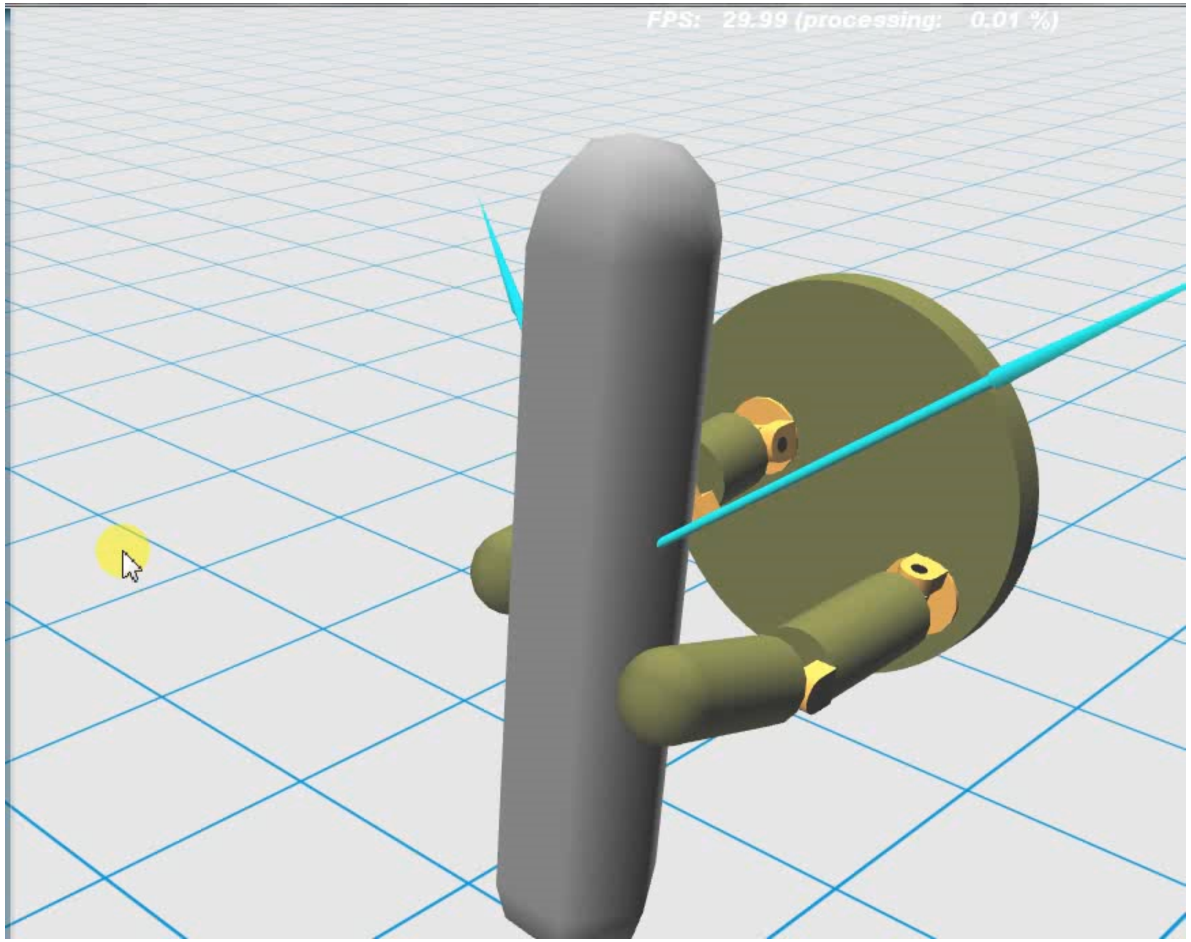


Tabletop from the side

Building Up a Motion From Primitives

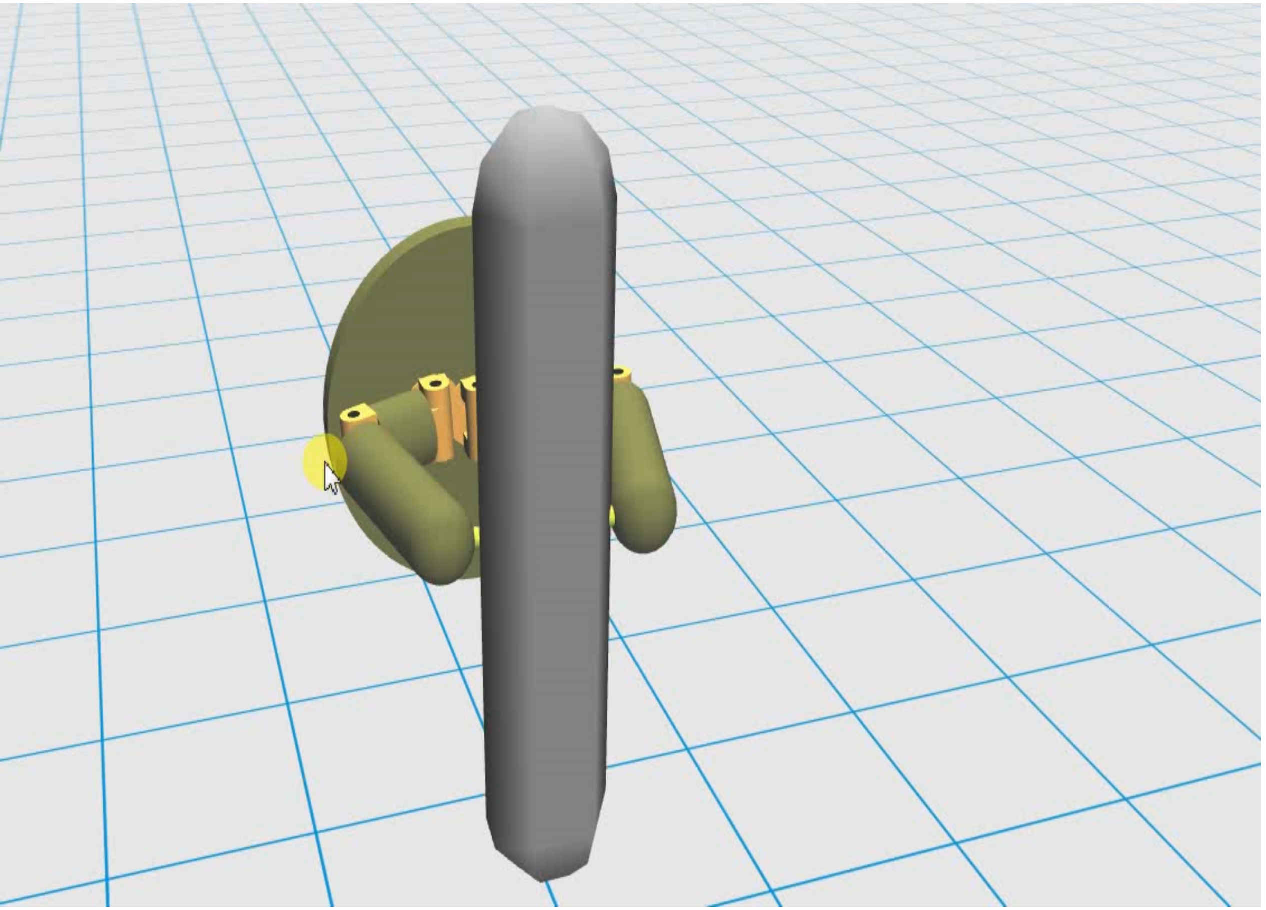


Horizontal (no gravity)

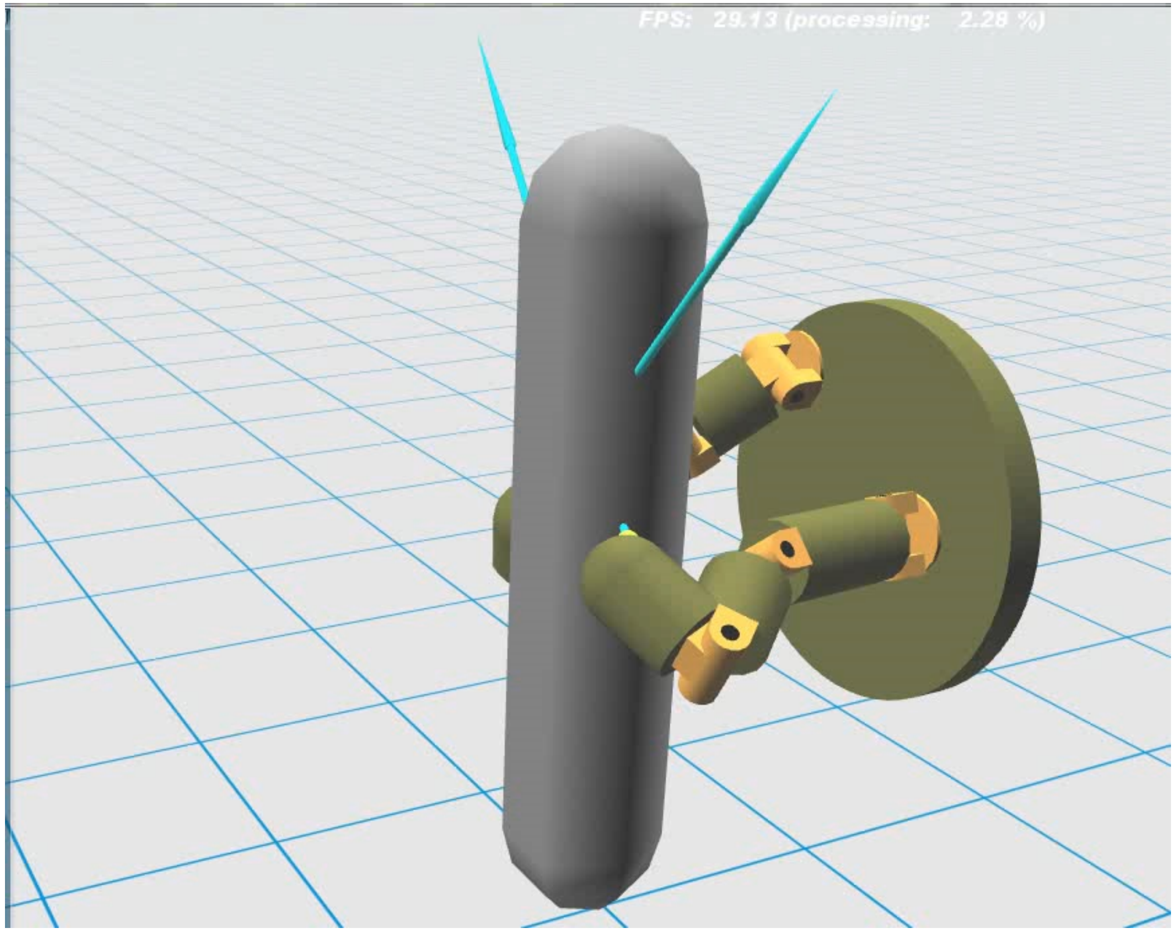


Horizontal (with gravity)

Building Up a Motion From Primitives

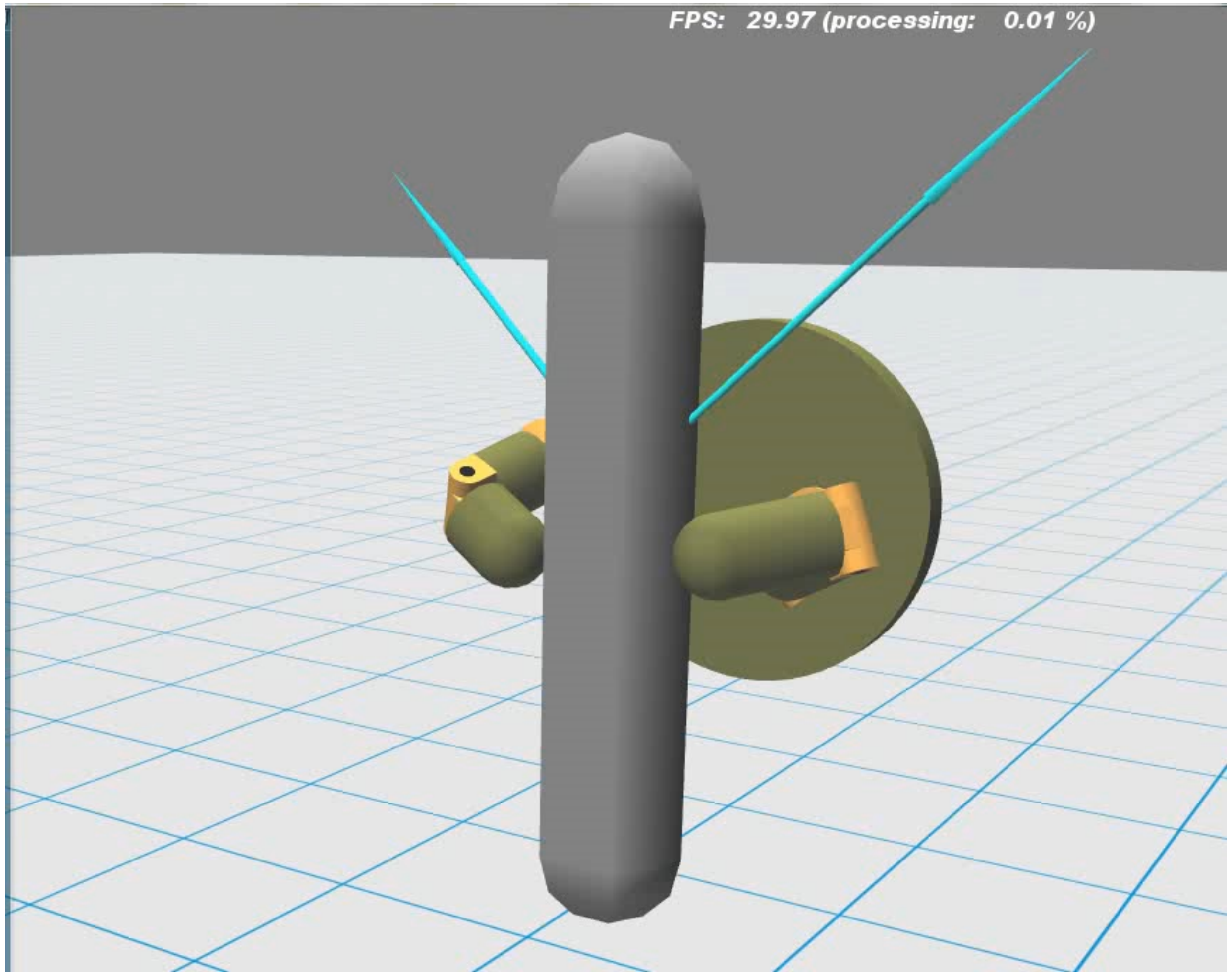


Circle in plane (no gravity)



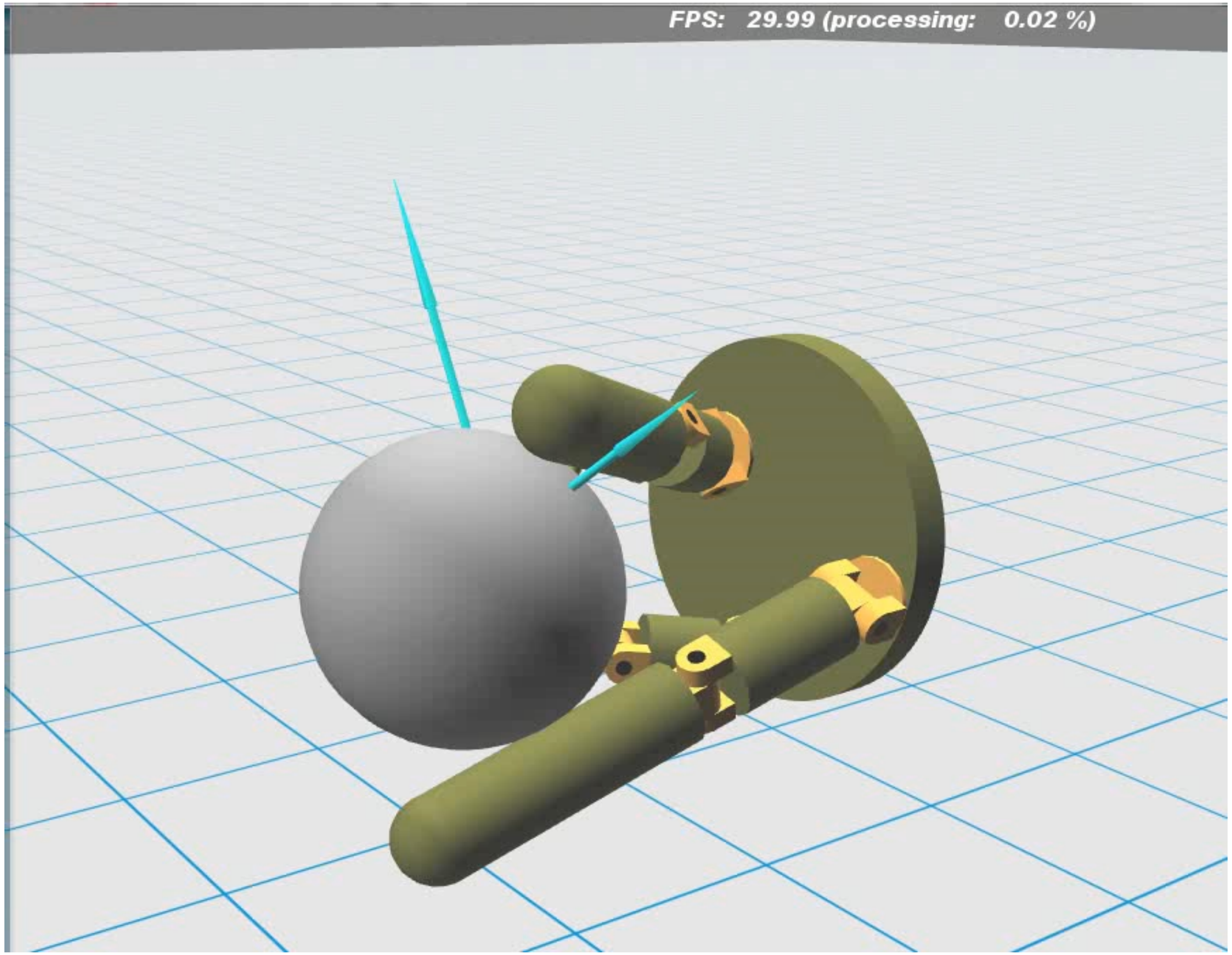
Circle in plane (with gravity)

Building Up a Motion From Primitives



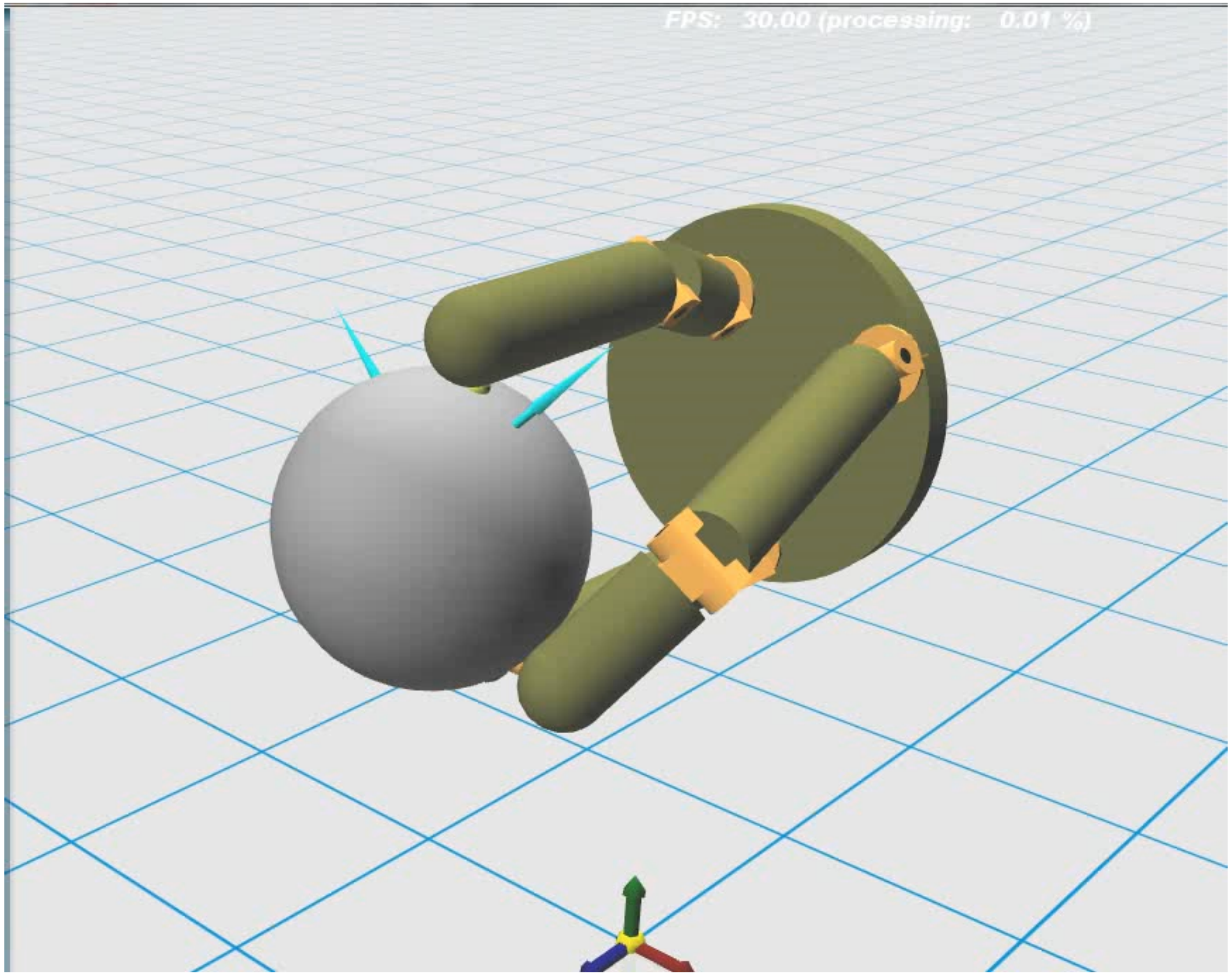
Hemisphere (with gravity)

“Multi-objective” Chaining Example



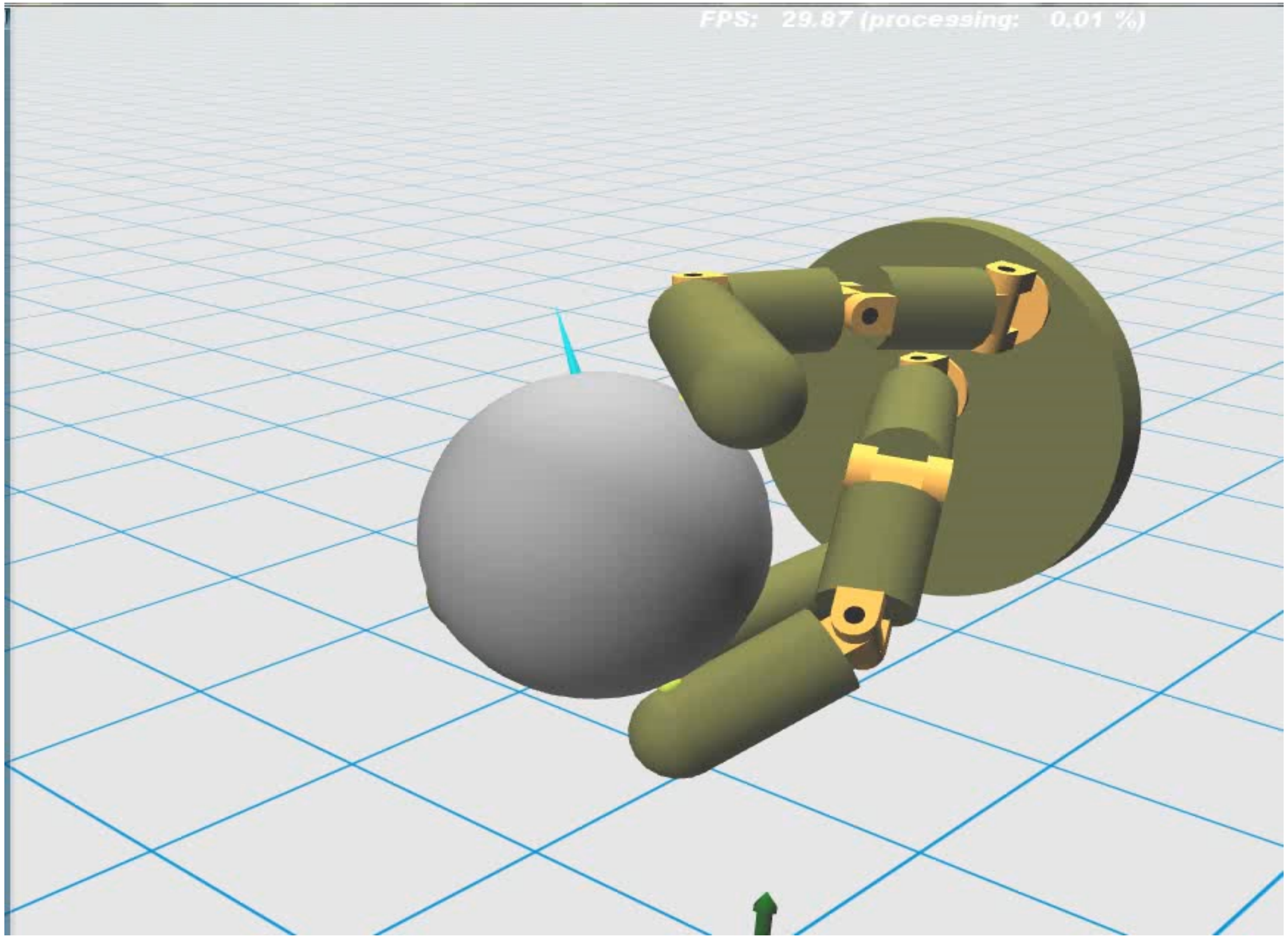
Sphere Rotation

“Multi-objective” Chaining Example



Sphere Rotation + translation

“Multi-objective” Chaining Example



Sphere Rotation + xy translation

Common Patterns

- The mechanisms for each task look totally different!
- Non-obvious/non-trivial designs
- Different numbers of links for each hand: scales with complexity
- Trajectory complexity tends to correspond to importance of fingers
- Hands become more aesthetically pleasing as we add more complexity to motion

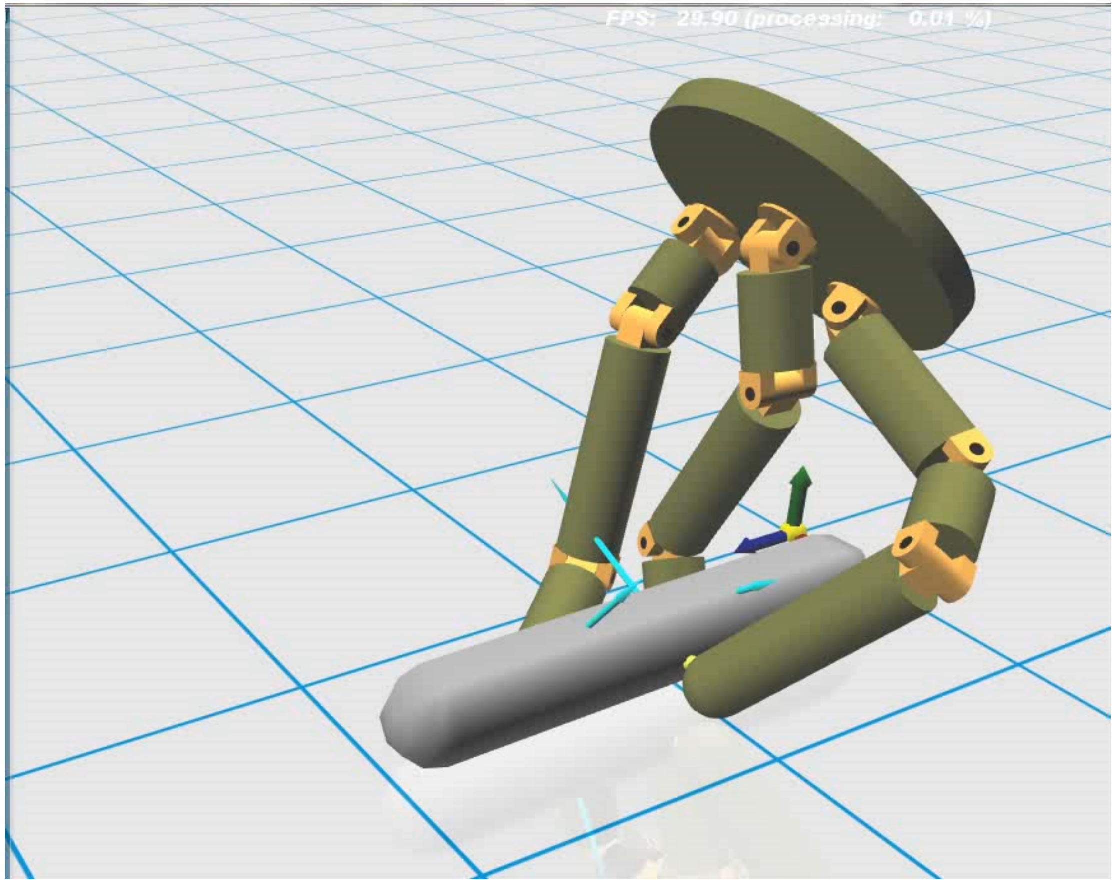
Limitations

- Slippage dynamics not exact
  -discouraged, not prohibited
  -usually not problematic except at high curvature

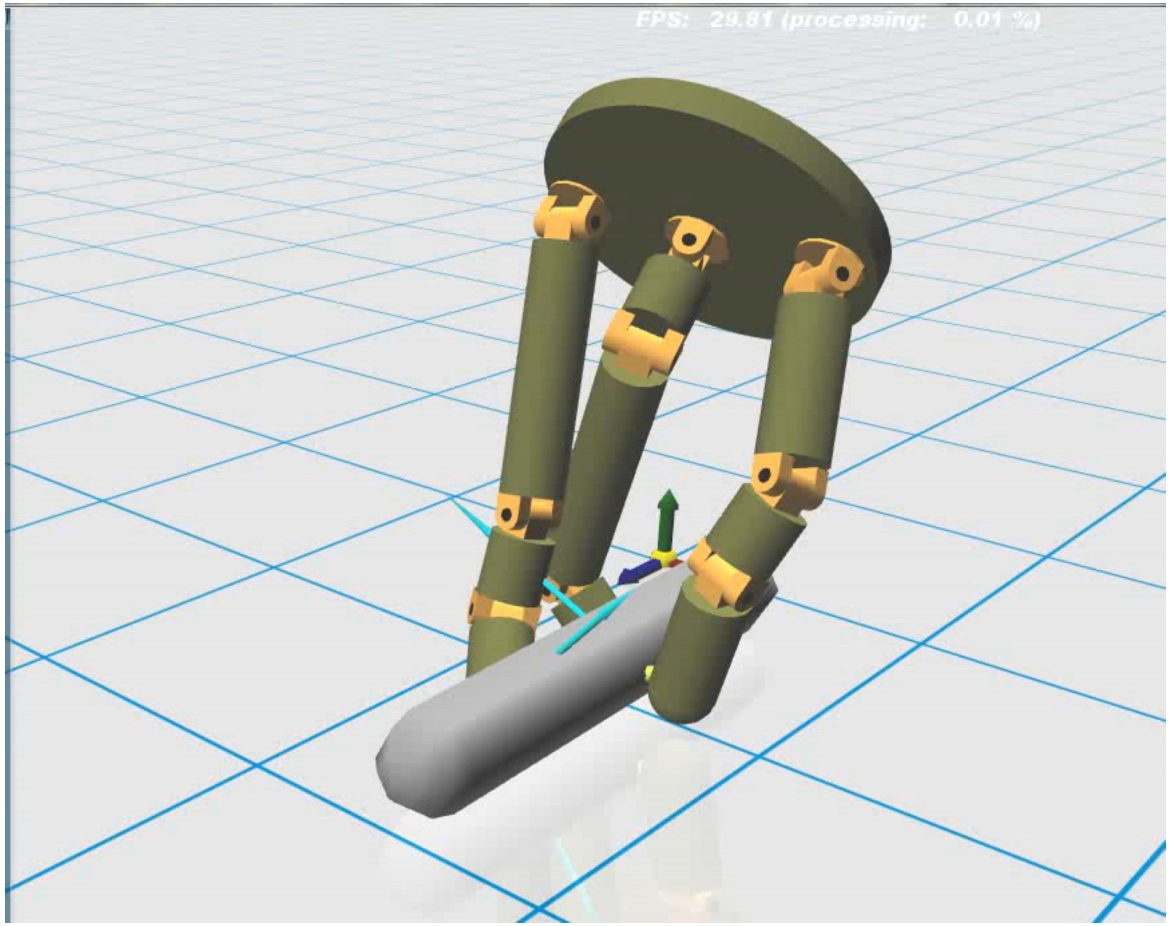
- User must provide a good base position and reasonable initial contacts
  - contacts selected with concept of fingers in mind

- Random contact initialization:
  -can work but unreliable
  -disconnect between optimization steps

Pencil Pickup Slip Demonstration

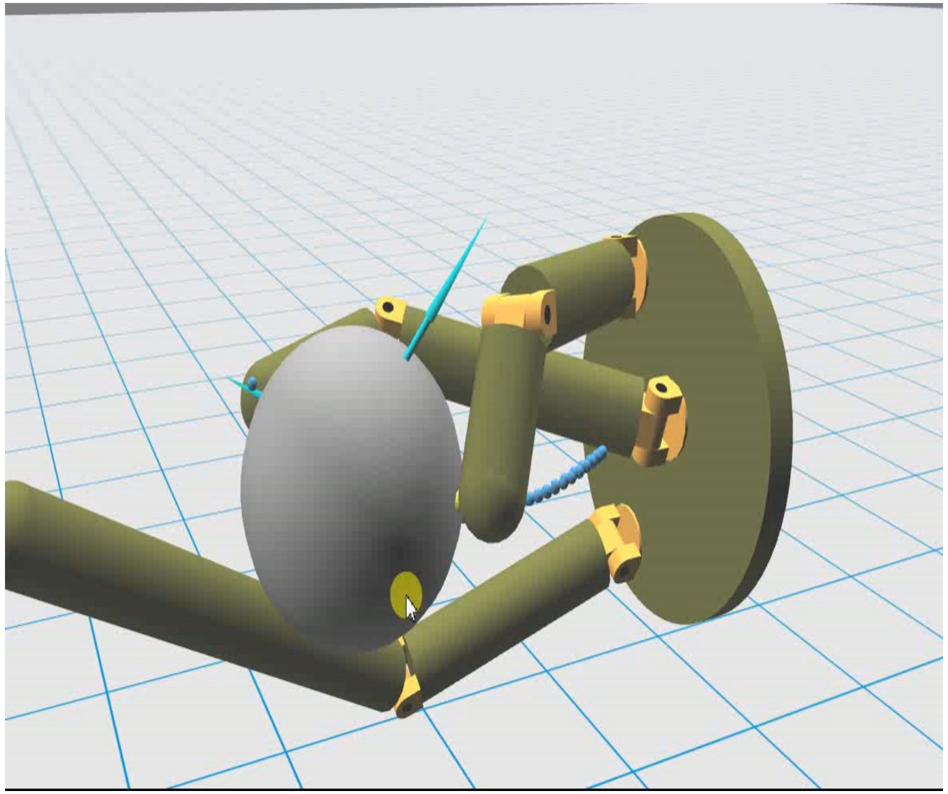


Acceptable Slippage

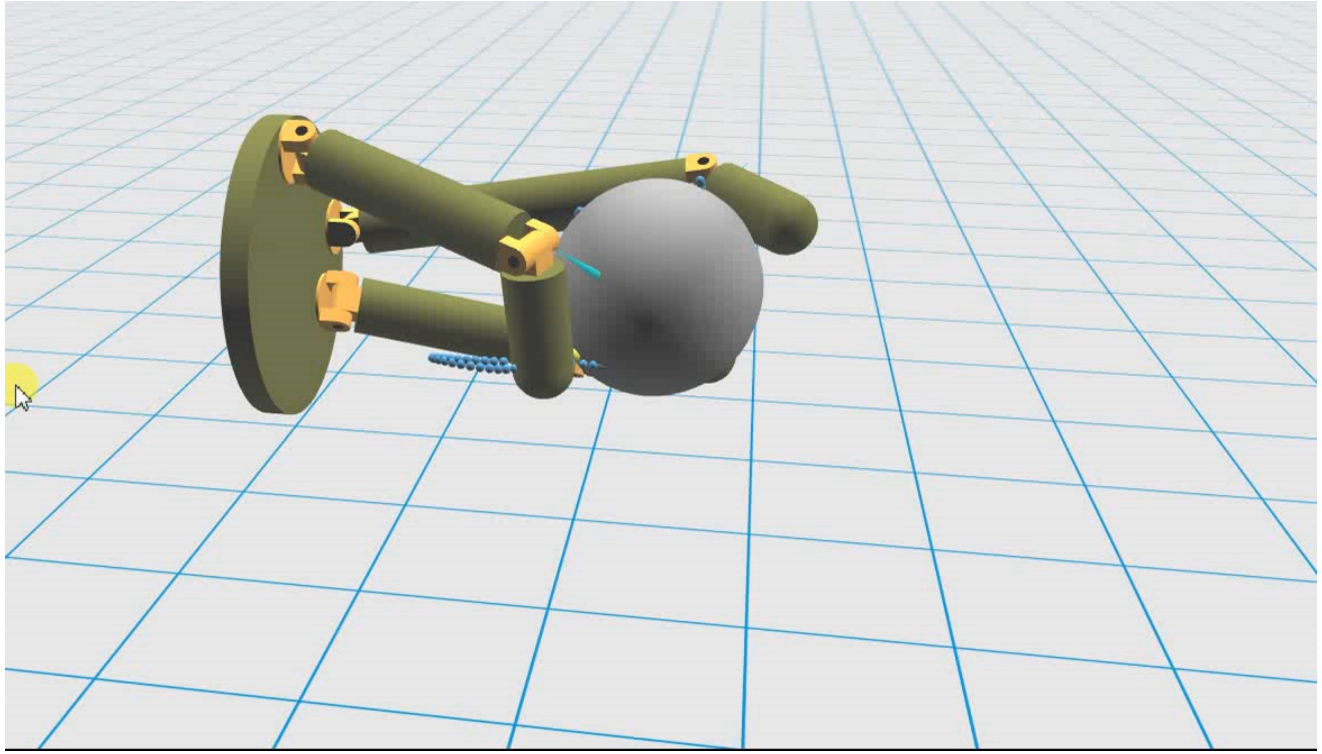


Uncomfortable Slippage

Automatic Contact Brittleness Demo



Sphere translate: Ok mechanism



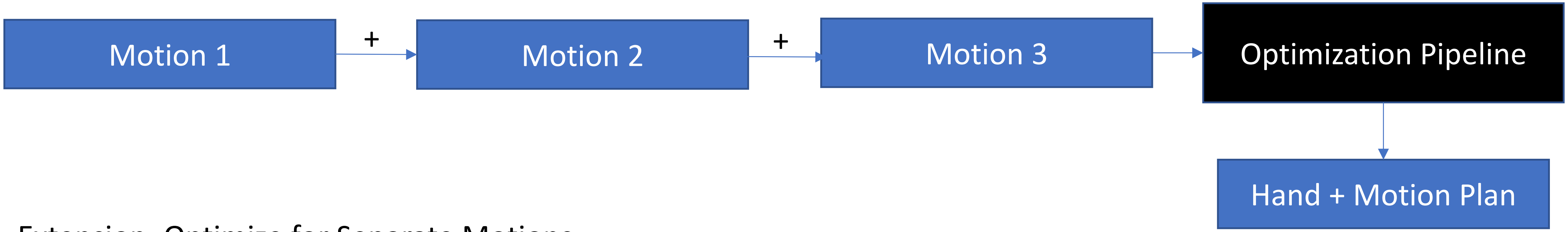
Sphere translate: Brittle mechanism

Additional Topics For The Future

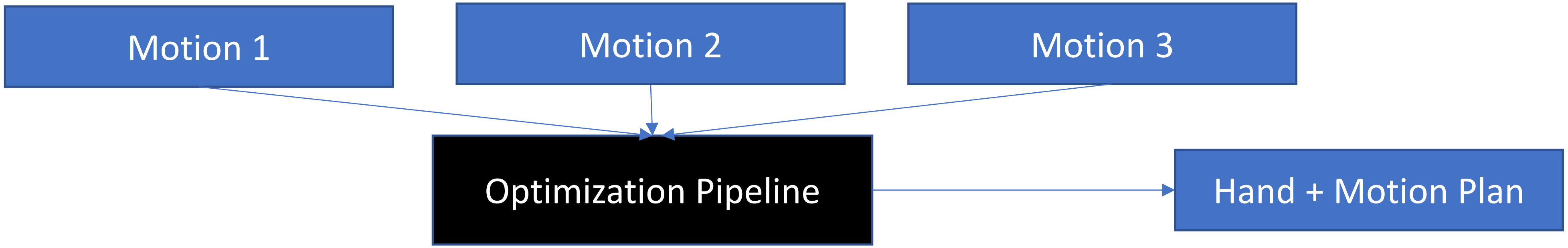
- Multi-objective optimization
- Initial Contact Planning
- Robustness through Physical Simulation
- Incorporating Dimensionality Reduction (Linkages/Synergies)

Multi-Objective Optimization

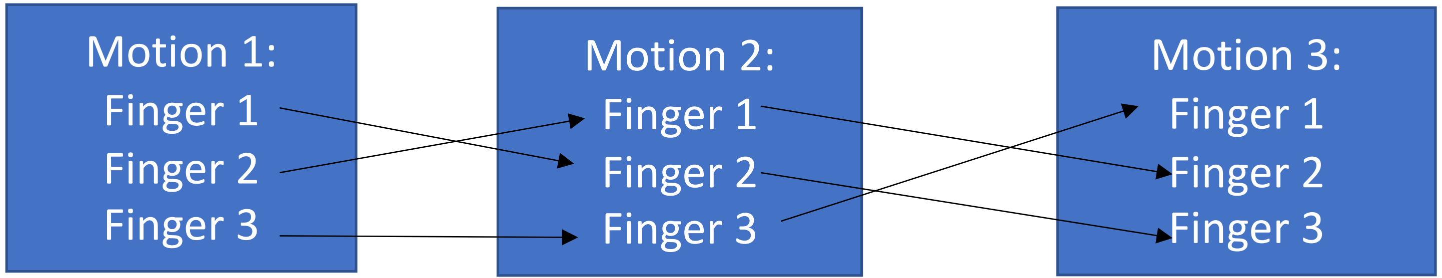
Current Capability: Motion Chaining



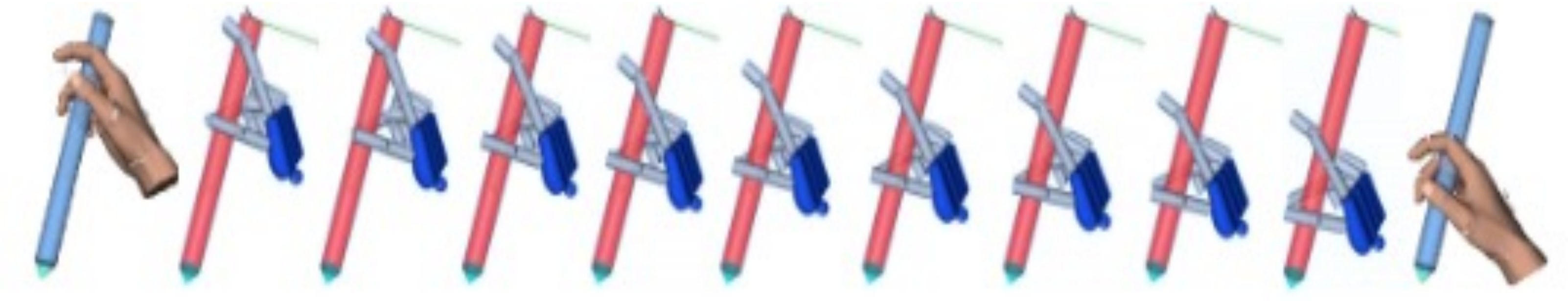
Extension: Optimize for Separate Motions



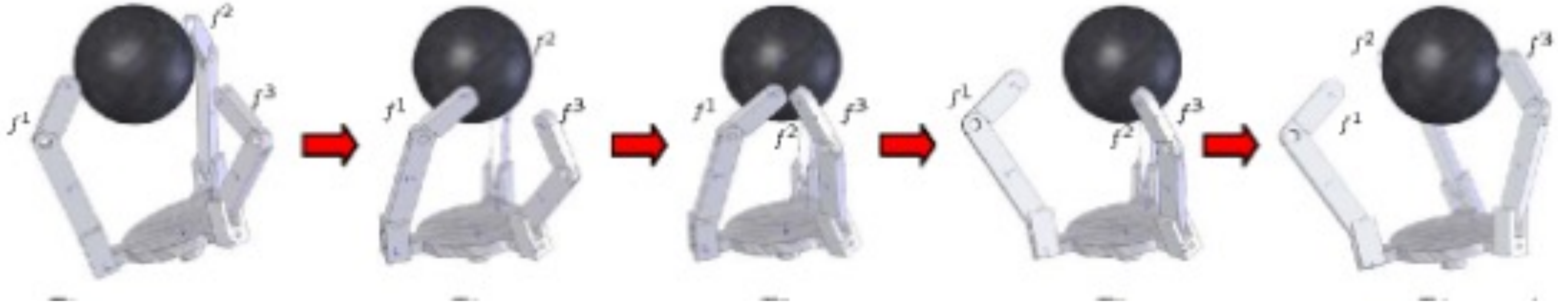
Problem: how do floating contacts match up?
---- $(n!)^{k-1}$ combos for n fingers, k motions



Initial Contact Planning/Additional Floating Heuristics



Vinayavekhin 2011: re-grasp on a cylinder



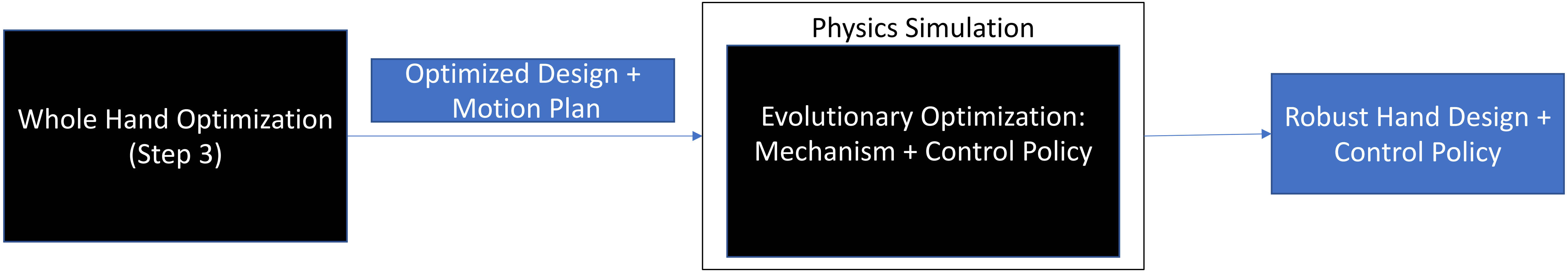
Xu 2007: finger gaiting for sphere rotation

Twirling a pencil

[1] Finger gaits planning for multifingered manipulation Xu 2007

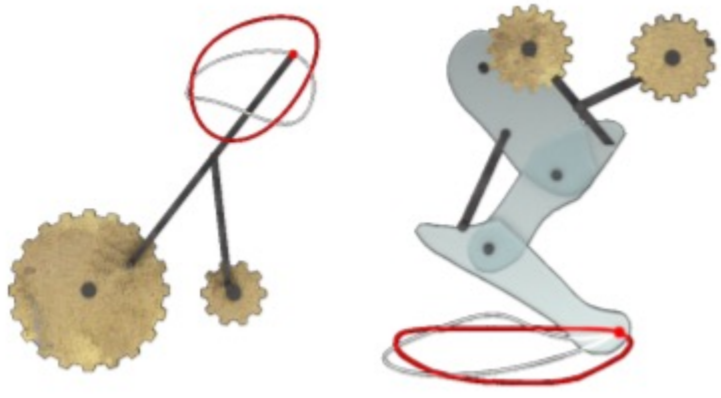
[2] Towards an automatic robot regrasping movement … Vinayavekhin 2011

Robustness Through Physical Simulation

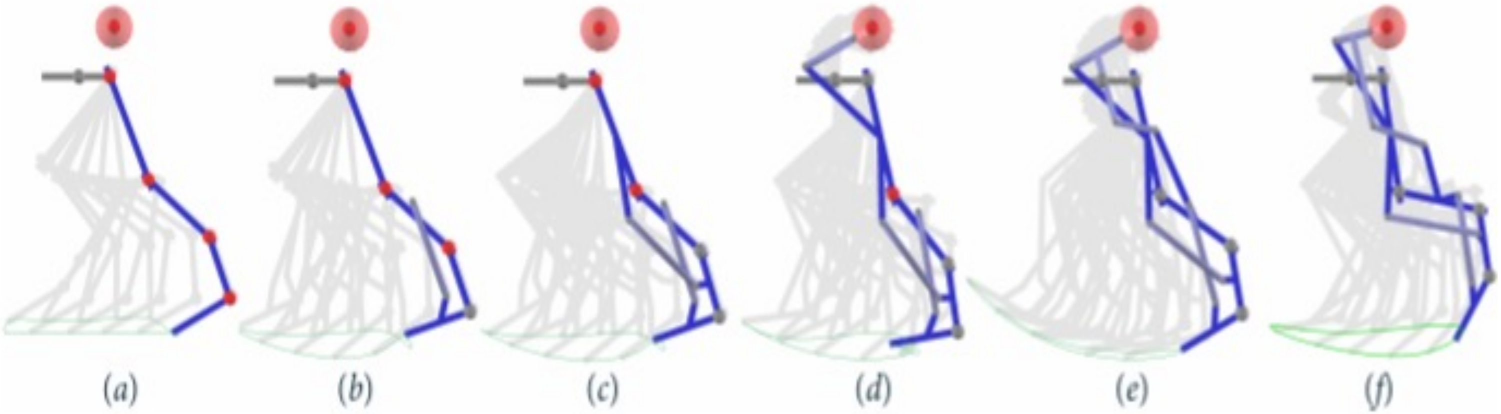


- Use synthesized mechanism as seed
- Control policy (torque): force feedback or open loop
- Gradient free optimization (e.g. Covariance Matrix Adaptation)
- Final step before fabrication

Dimensionality Reduction



Coros 2013: linkage tracks target curve (red)



Thomaszewski 2014: motor replacement steps

Coros 2013: design and fabrication example

[1] "Computational design of mechanical characters" Coros 2013

[2] “Computational design of linkage-based characters” Thomaszewski 2014