Tree-Based Representations of Variational Integrators

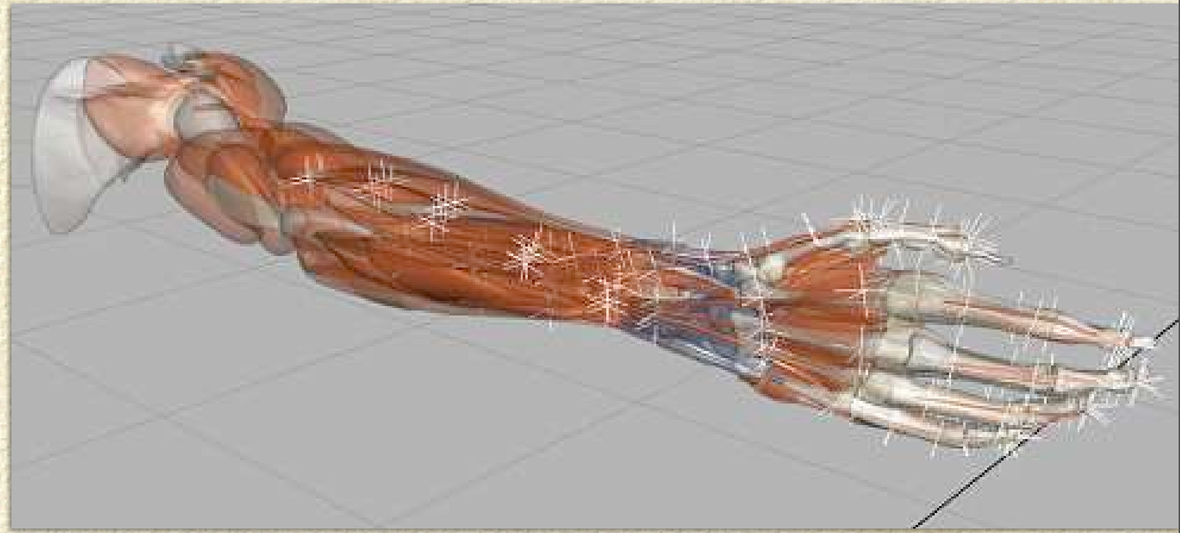
Todd Murphey
Northwestern University
May 8, 2009

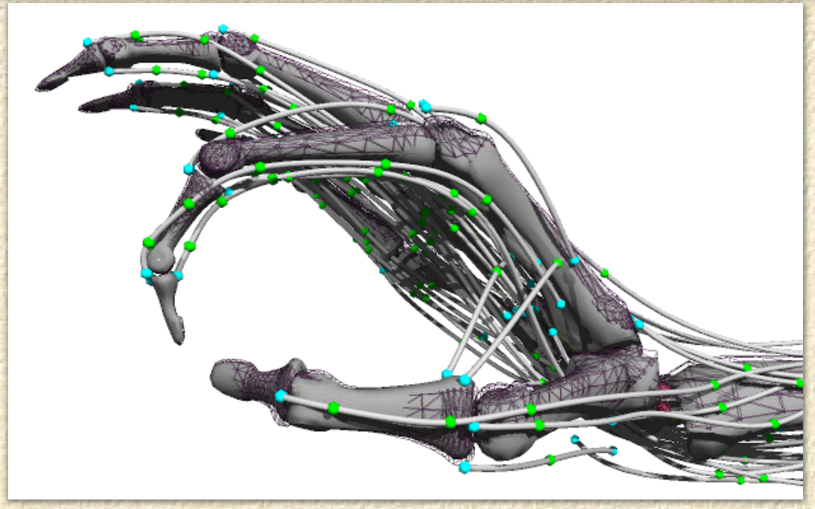
Engineering and Complexity

- Software matters
  - need to deal with complexity management
    - *M(q)* is not typically constant in general coordinates
  - need correct specification
  - Often need *real-time* simulation
  - The CS fast simulation techniques are popular because they do these things
- Variational Integration's major advantage--configurations
  - motion planning and other CS disciplines
  - estimation
  - system identification

Complex Systems: Hand Dynamics

- Tissue mechanics is well-developed for analyzing individual tissues
- No full-hand numerical models currently exist
- not an engineered system so very noncollocated
- "Strands" provide a finite-dimensional representation that is amenable to analysis

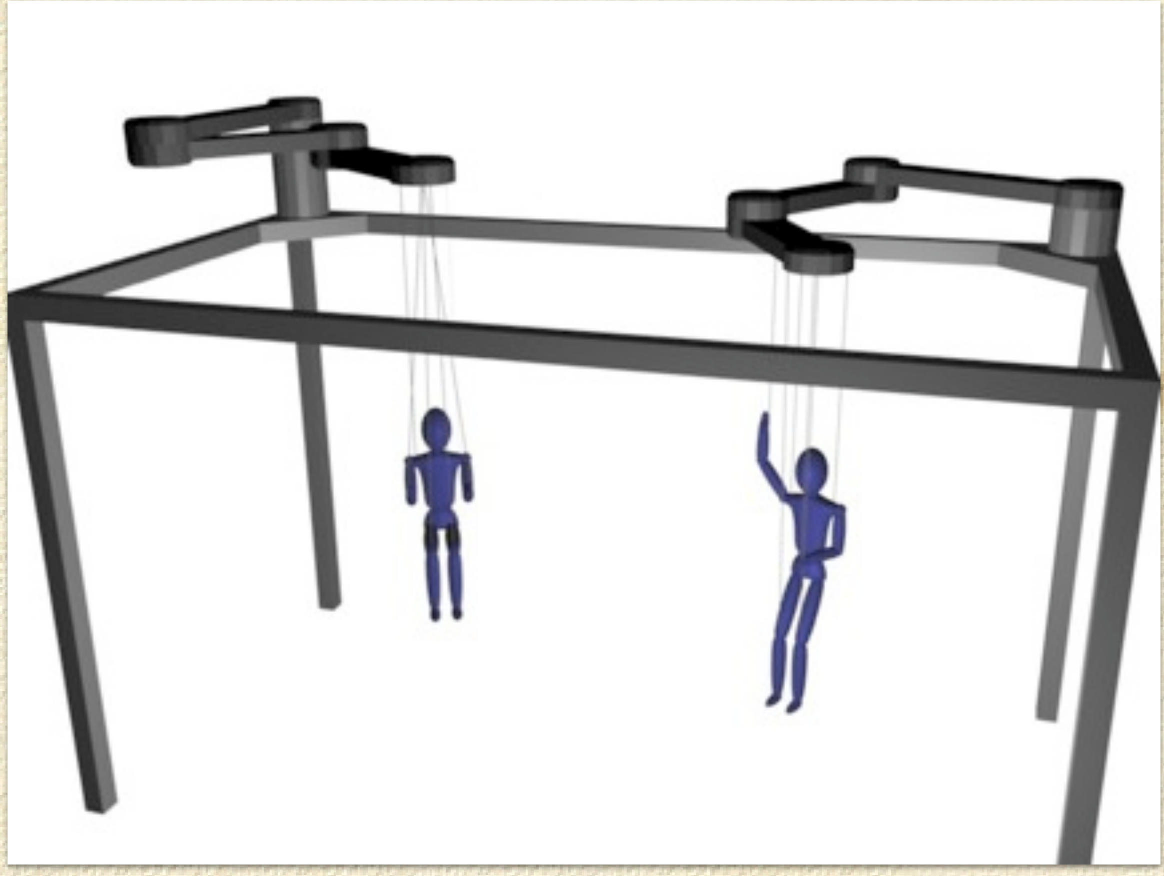




Pai et al, SIGGRAPH 2008

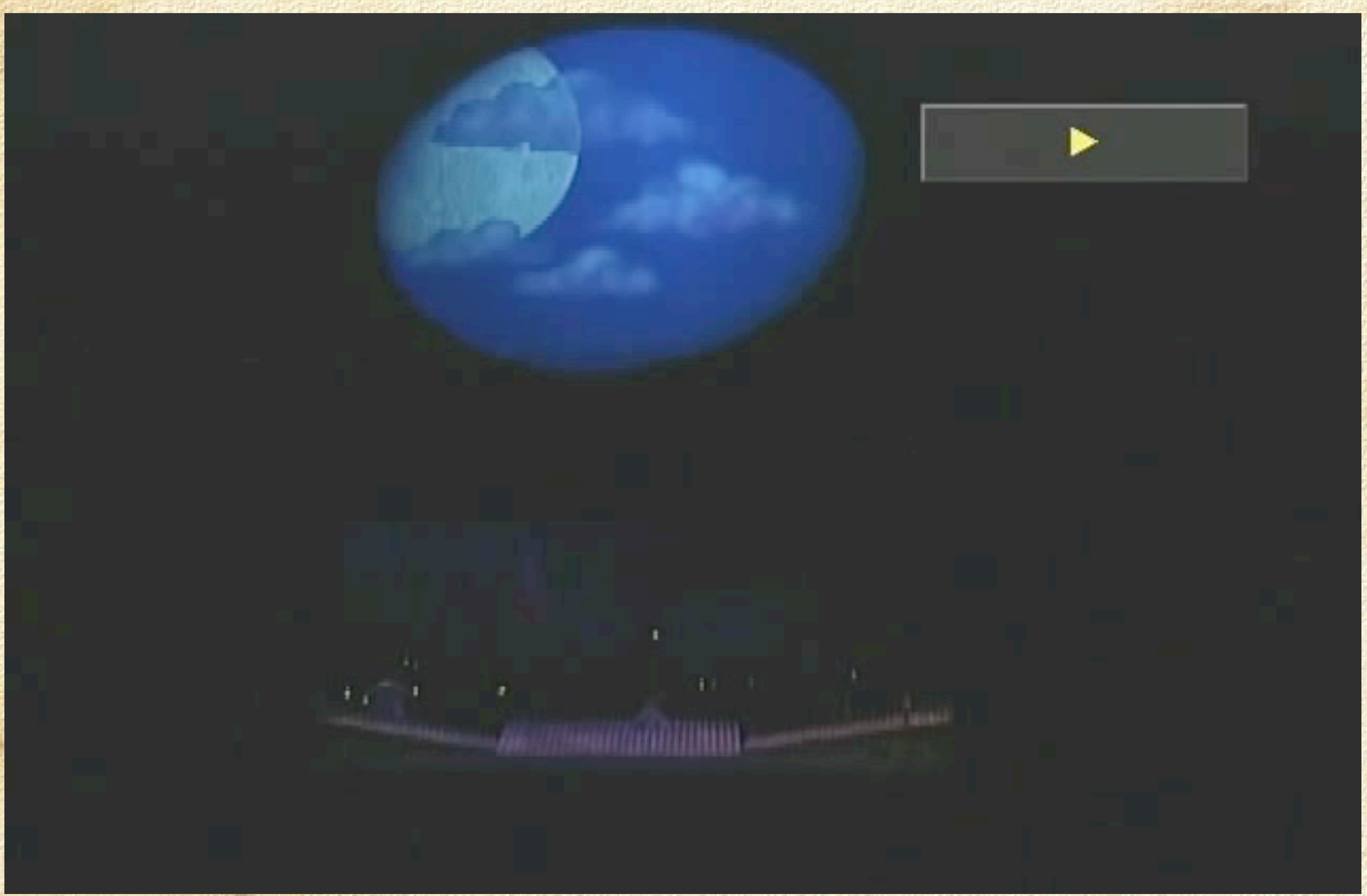
Complex Systems: String Puppets

- 40-50 DOF
- Nontrivial constraints
- Generalized coordinates for control analysis
- Force balance by hand is not feasible
- closed kinematic chains
- Control is ... difficult
- Motion Description Languages (MDLs)



Collaboration with Magnus Egerstedt at Georgia Tech, Atlanta Center for Puppetry Arts, and Disney R&D/Imagineering

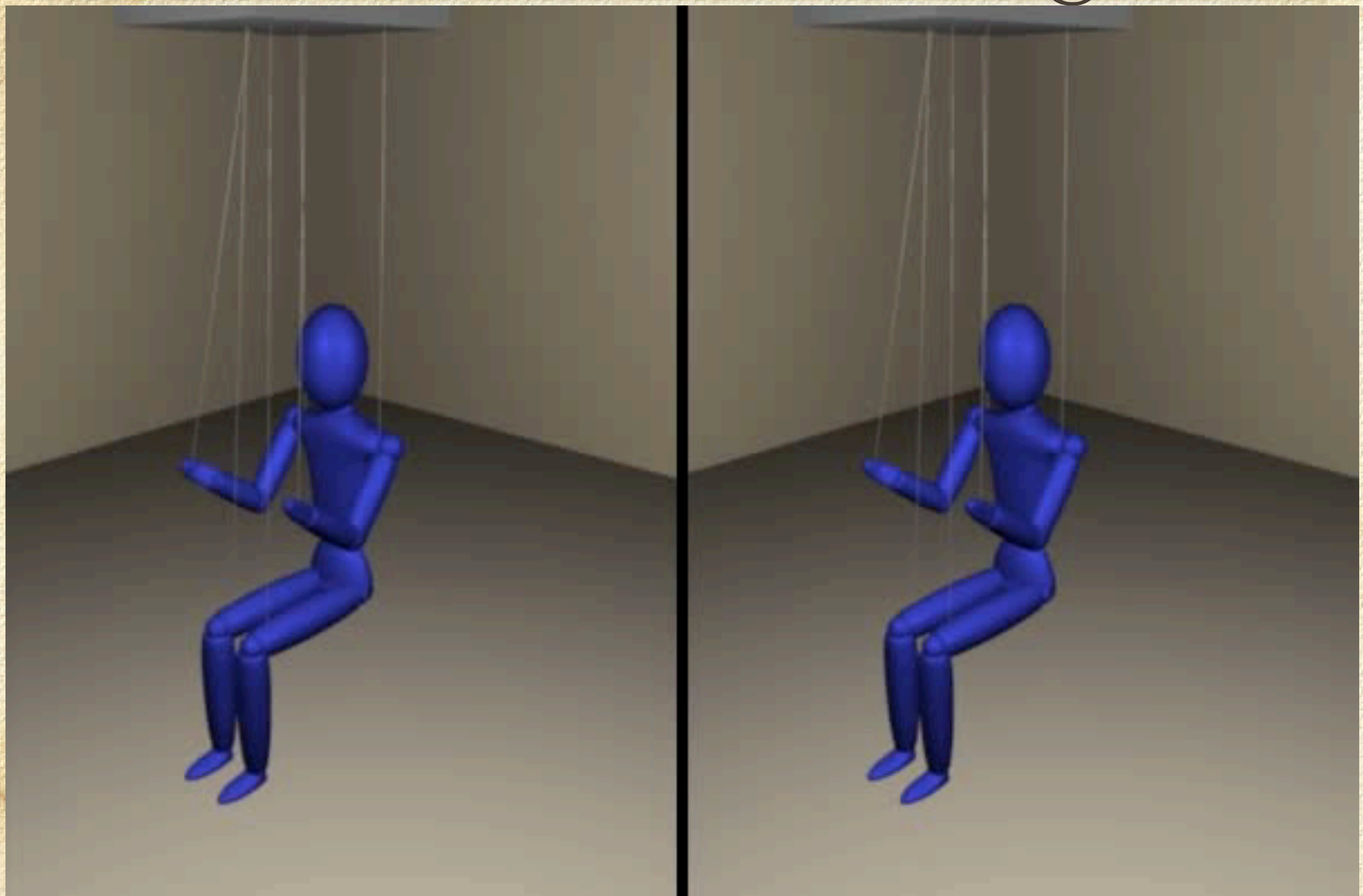
Puppeteers Solve the Complexity Problem



Avanti Da Vinci--Atlanta Center Puppetry Arts 2005.

Aside: Need for Visualization

Which One is Wrong?



Need reliable constrained mechanical simulations

Simulation Needs

- Speed required for multiple simulations for:
  - probabilistic planning
  - optimization
- Good convergence properties
- Good properties with respect to identification
- Must scale to high dimensional (100-1000 DOF) systems to be a useful tool in traditional engineering disciplines
- Algorithmic for arbitrary
  - numbers of rigid bodies
  - types of interconnections
- Avoid system-specific modeling and software architecture

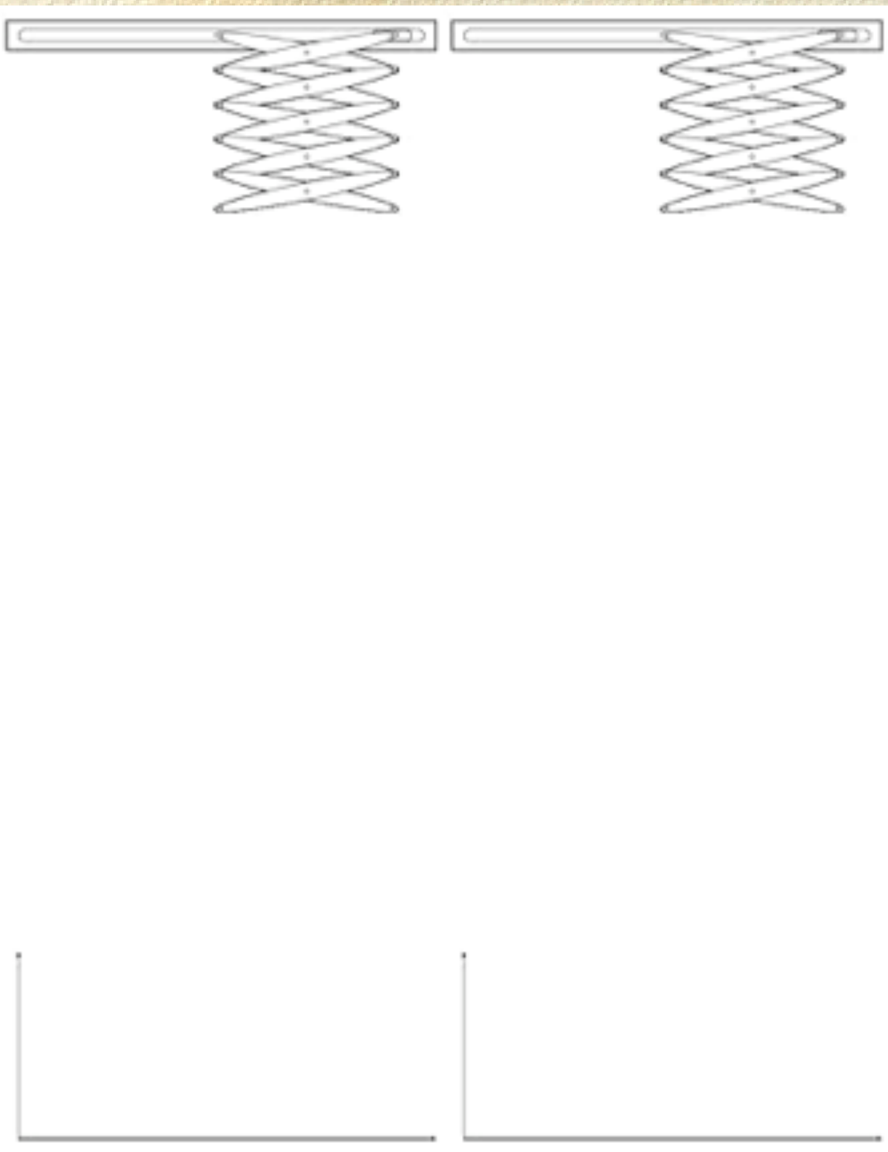
Good Simulation Exists

- Fast physics simulation tools already exist...
  - R. Featherstone, *Robot Dynamics Algorithms* (1987)
  - D. Baraff, many articles (1989-2003)
- Designed for animation/graphics applications
  - We use Open Dynamics Engine (ODE)
- Rigid bodies that interact with forces
- Handles constraints, forces, impacts...typically heuristic
- Specification using rigid body transformations
- Spurious behavior of simulation has been accepted
  - *"Numerical dissipation is fine because real systems have dissipation"*

Problems with System Identification

- System identification requires simulation
- Rigid body simulation can introduce significant artificial numerical dissipation, particularly for constrained systems
  - leads to qualitatively (and quantitatively) incorrect simulation
  - leads to potentially catastrophically bad system identification
- Lightly damped systems such as tendon networks are particularly susceptible to these problems

Example System Identification: Scissor Lift



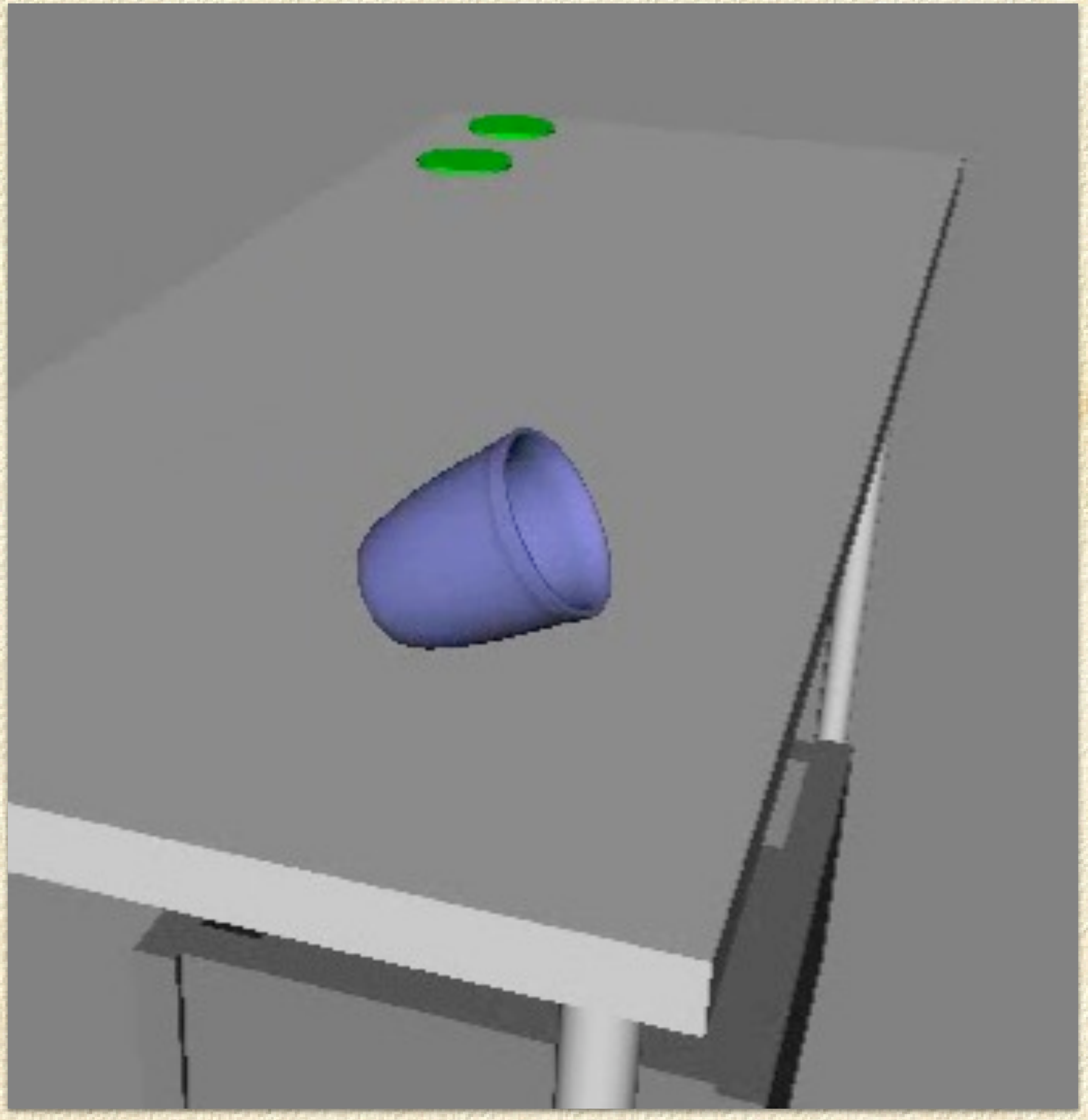
ODE *Real*

- 60 DOF system 59 constraints
- 1 DOF ODE exists
- Open Dynamics Engine (ODE) dissipates energy very quickly
- Suppose *real* system is not heavily damped,
- Linearized identification has RHP poles!

Open Dynamics Engine (ODE) with constraints

- Newton-Euler approach leads to nonphysical outcomes
- Closed kinematic chains are notoriously difficult to simulate
- We use ODE in our benchmark tests

Movie courtesy Siddhartha Srinivasa and Dmitry Berenson at Intel Robotics R&D



Discrete Euler-Lagrange Equations

$$D_2 L_d(q_{i-1}, q_i) + D_1 L_d(q_i, q_{i+1}) = -\lambda_i \nabla h(q_i)$$

$$h(q_{i+1}) = 0$$

- Given q_{i-1} and q_i, solve for q_{i+1} (root-finding problem)
- Directly approximating the Least Action Principle, (***not*** an ODE), and can use any quadrature rule
- Excellent energy, momentum, and convergence behavior
- Resulting trajectory approximates the actual solution to same order as discrete L_d approximates actual L

Discrete Euler-Lagrange Equations

- Note that there are *no* velocities in this formulation--only discrete states
- Does not require all interactions to be between rigid bodies (key for simulating strand dynamics)
- Algorithmically expressing variational integrators requires an approach similar to Featherstone's work; we recursively evaluate all of the terms in

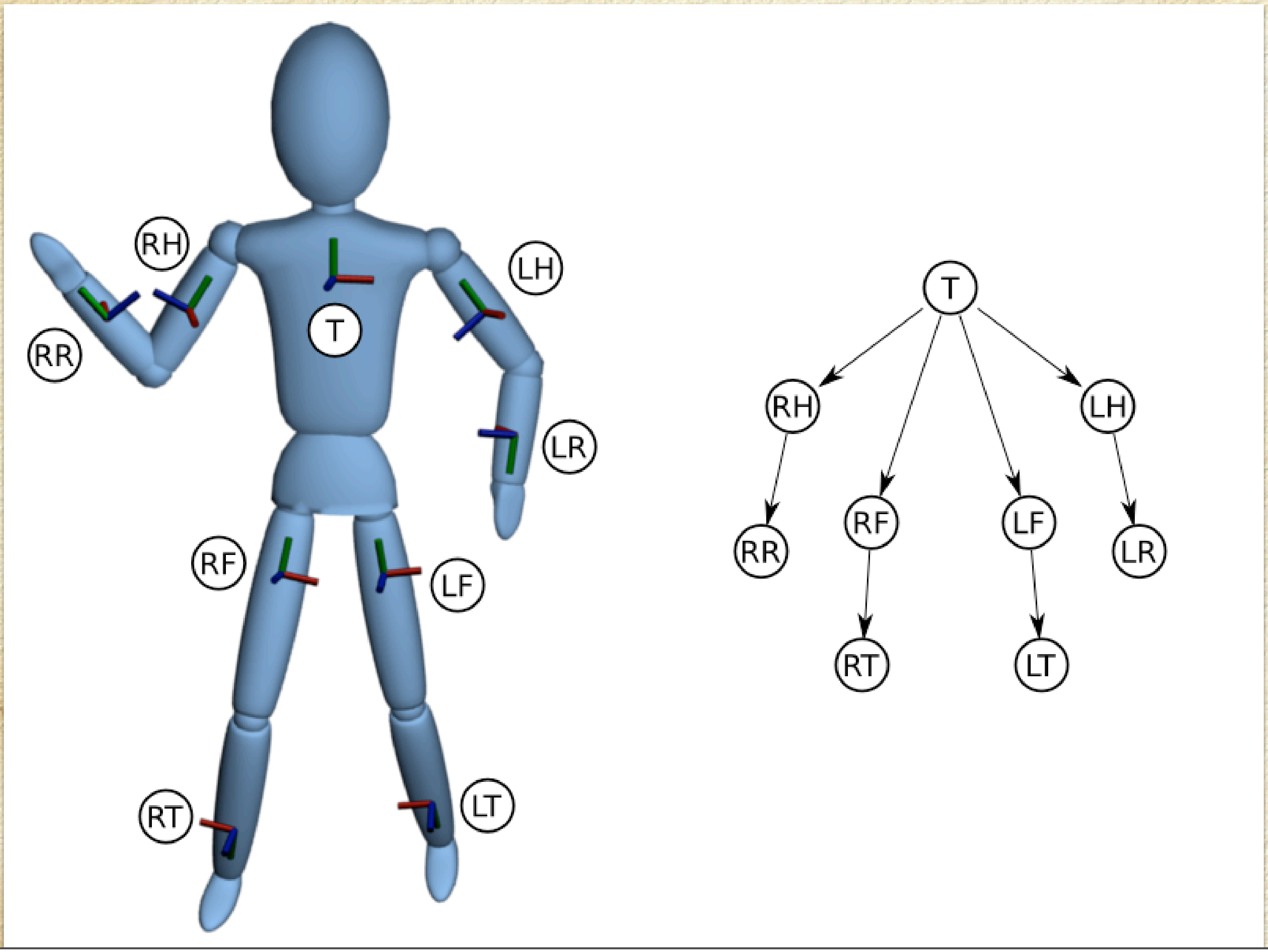
$$D_2 L_d(q_{i-1}, q_i) + D_1 L_d(q_i, q_{i+1}) = -\lambda_i \nabla h(q_i)$$

$$h(q_{i+1}) = 0$$

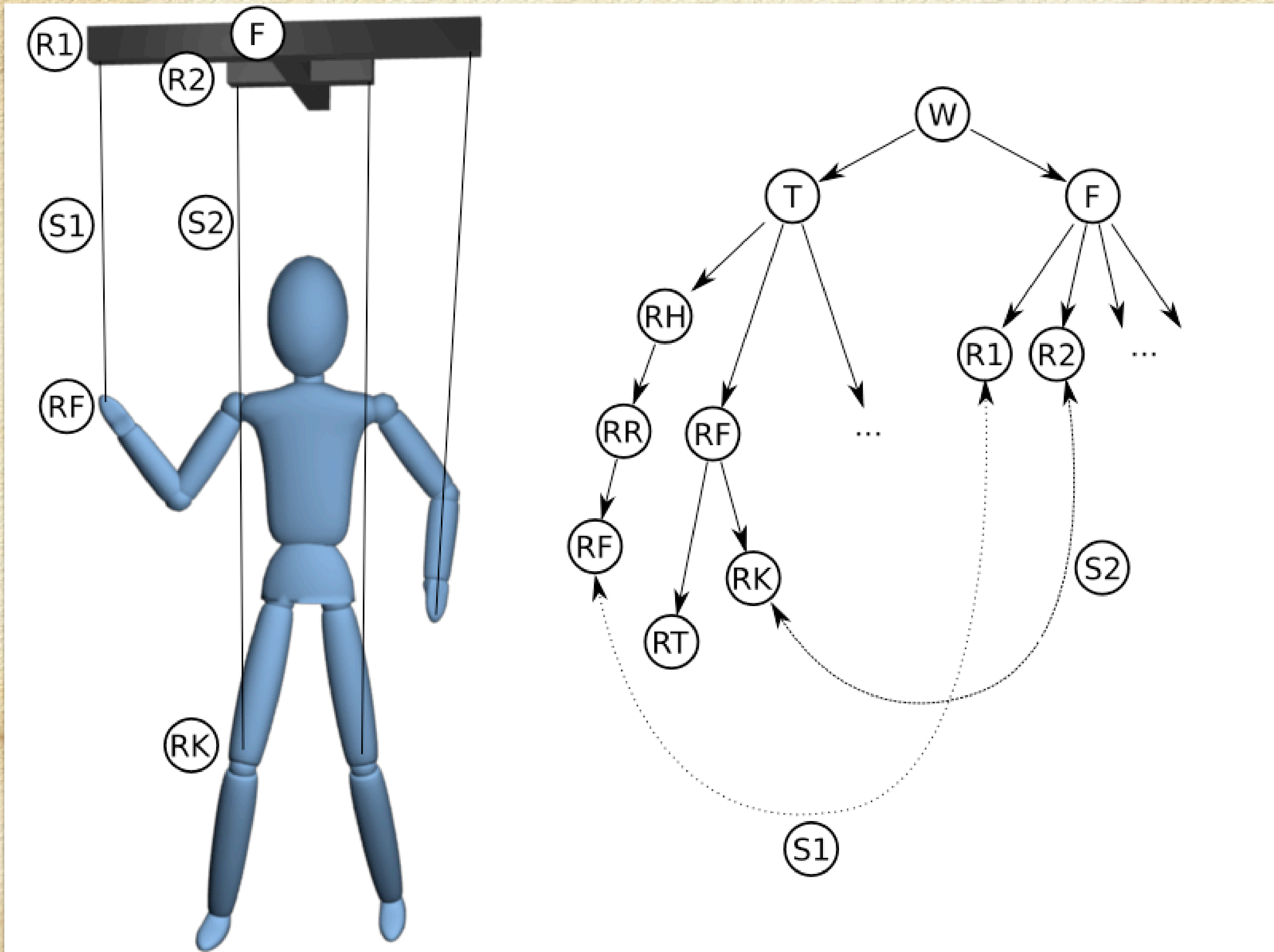
Scalability: Graph Representations for Complex Systems

- Graphs provide a way of organizing information
- Algebraic Graph G, with edges E and vertices V
- Vertices V correspond to individual bodies
  (inertia tensors and external forces acting on the body)
- Edges E are relative transformations between bodies
  (rigid body transformations and constraints)

Example: Graph Representation



Example: Constrained Systems



Computing $D_2 L_d(q_{i-1}, q_i) + D_1 L_d(q_i, q_{i+1}) = 0$

- Recursively evaluate derivative information

$$g_k = \begin{bmatrix} R_k & p_k \\ 0 & 1 \end{bmatrix}$$

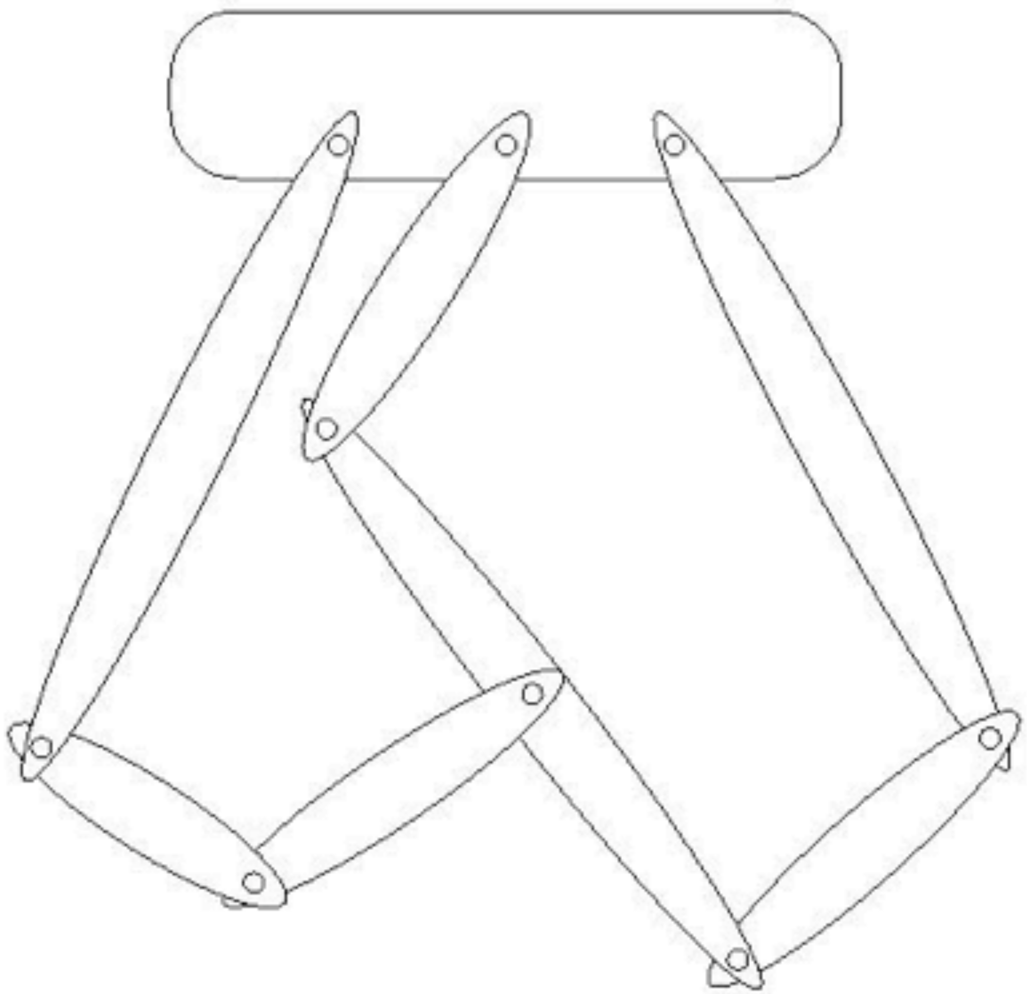
$$\hat{v}_k = g_k^{-1} \dot{g}_k$$

$$\frac{\partial}{\partial q_i} \hat{v}_k^b(q, \dot{q}) = \begin{cases} 0 & k = s \\ & i \notin \mathrm{anc}(k) \\ g_k^{-1\prime} \hat{v}^b_{\mathrm{par}(k)} g_k + g_k^{-1} \hat{v}^b_{\mathrm{par}(k)} g_k' & i = k \\ g_k^{-1} \frac{\partial}{\partial q_i} \hat{v}^b_{\mathrm{par}(k)} g_k & i \neq k \end{cases}$$

- Root solving requires the linearization as well
- Arbitrary order approximations can be calculated--linearization, etcetera
- Caching and Parallelization vital

Johnson and Murphey, IEEE Transactions on Robotics 2008, Accepted

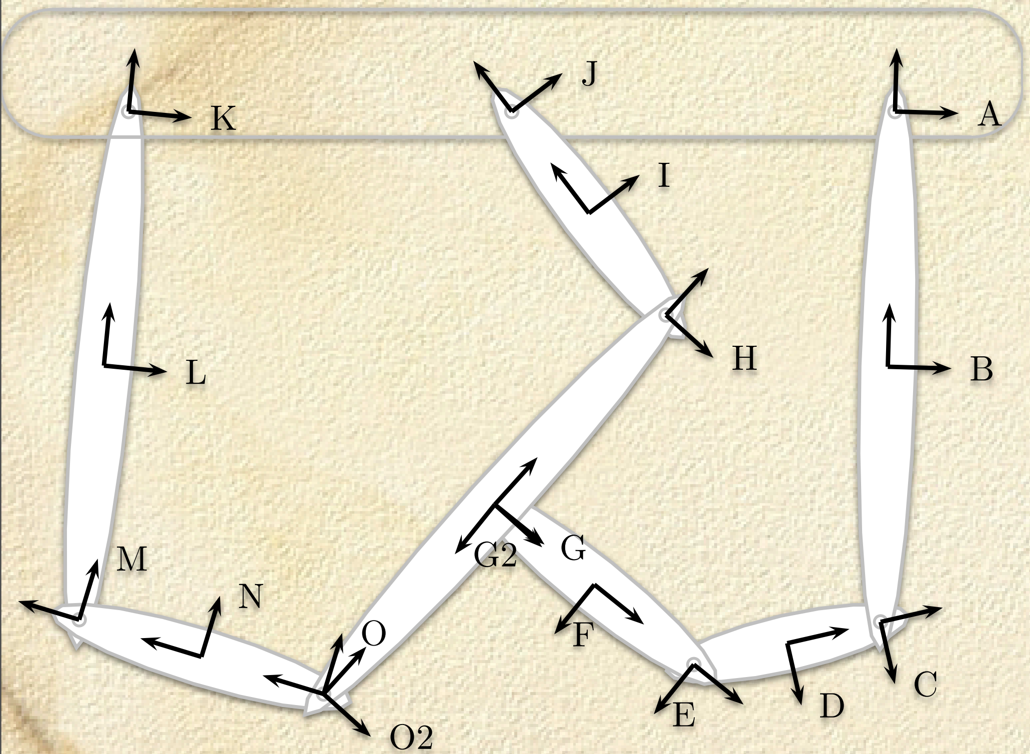
Kinematic Closed Chains



Nakamura and Yamane. IEEE Transactions on Robotics and Automation, 16(2), 2000.

- Seven DOF
- Two constraints
- Time step 0.01 s
- 3 s for 10 s simulation
- Iterations/step: ~3.08
- Tolerance: 10^{-10}
- Often times *one* step is all that is necessary!

Code: **TREP**




```
(mechanical-system (gravity 0 0 -9.81)
  (ry "J" (Name "J")
      (tz -0.5 (Name "I") (Mass 1))
      (tz -1.0
          (ry "H" (Name "H")
              (tz -1.0 (Name "G") (Mass 1))
              (tz -2.0 (Name "O2")))))
  (tx -1.5
      (ry "K" (Name "K")
          (tz -1.0 (Name "L") (Mass 1))
          (tz -2.0
              (ry "M" (Name "M")
                  (tz -0.5 (Name "N") (Mass 1))
                  (tz -1.0 (Name "O"))))))

  (tx 1.5
      (ry "A" (Name "A")
          (tz -1.0 (Name "B") (Mass 1))
          (tz -2.0
              (ry "C" (Name "C")
                  (tz -0.375 (Name "D") (Mass 1))
                  (tz -0.75
                      (ry "E" (Name "E")
                          (tz -0.5 (Name "F") (Mass 1))
                          (tz -1.0 (Name "G2"))))))))
 (point-constaint "G" "G2" (1 0 0))
 (point-constaint "G" "G2" (0 0 1))
 (point-constaint "O" "O2" (1 0 0))
 (point-constaint "O" "O2" (0 0 1)))
```

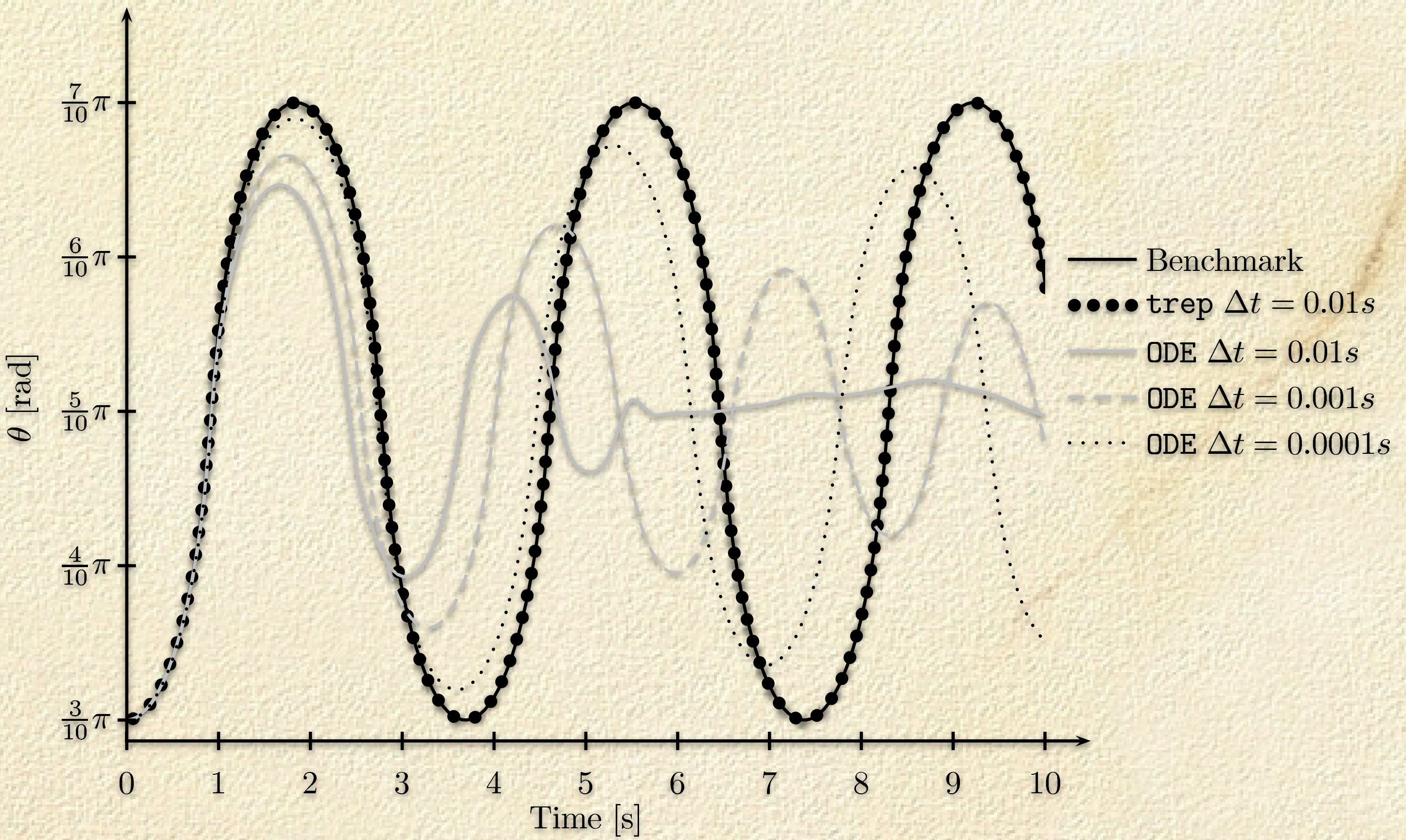
Transformations between frames generate system specification

Goal: MATLAB for nonlinear mechanical control systems

This is literally the code used to simulate the system

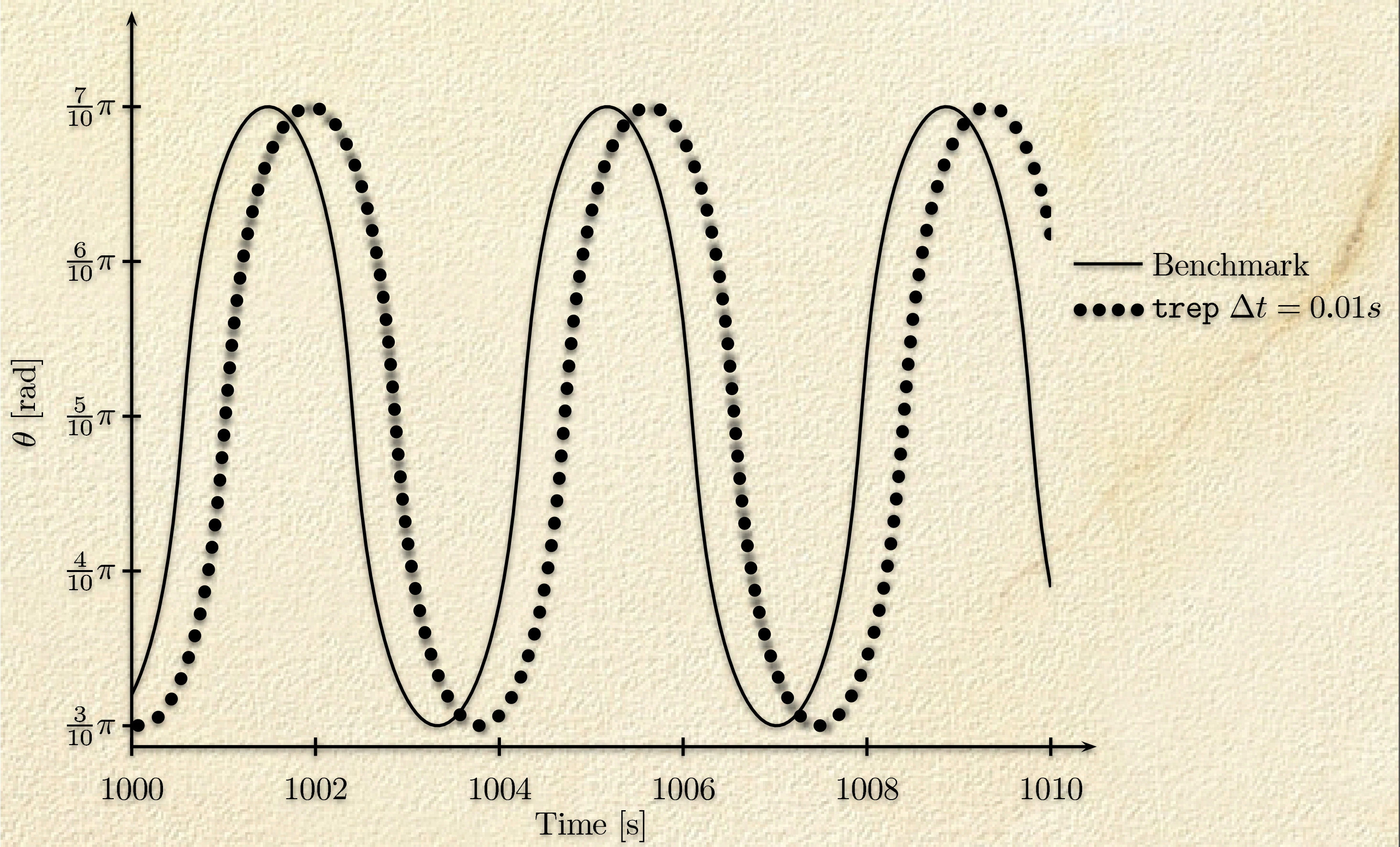
ODE comparison: A 1-dof/n-dof example

ODE vs variational integrator



0.01 steps in trep take about the same amount of time as 0.001 steps in ODE

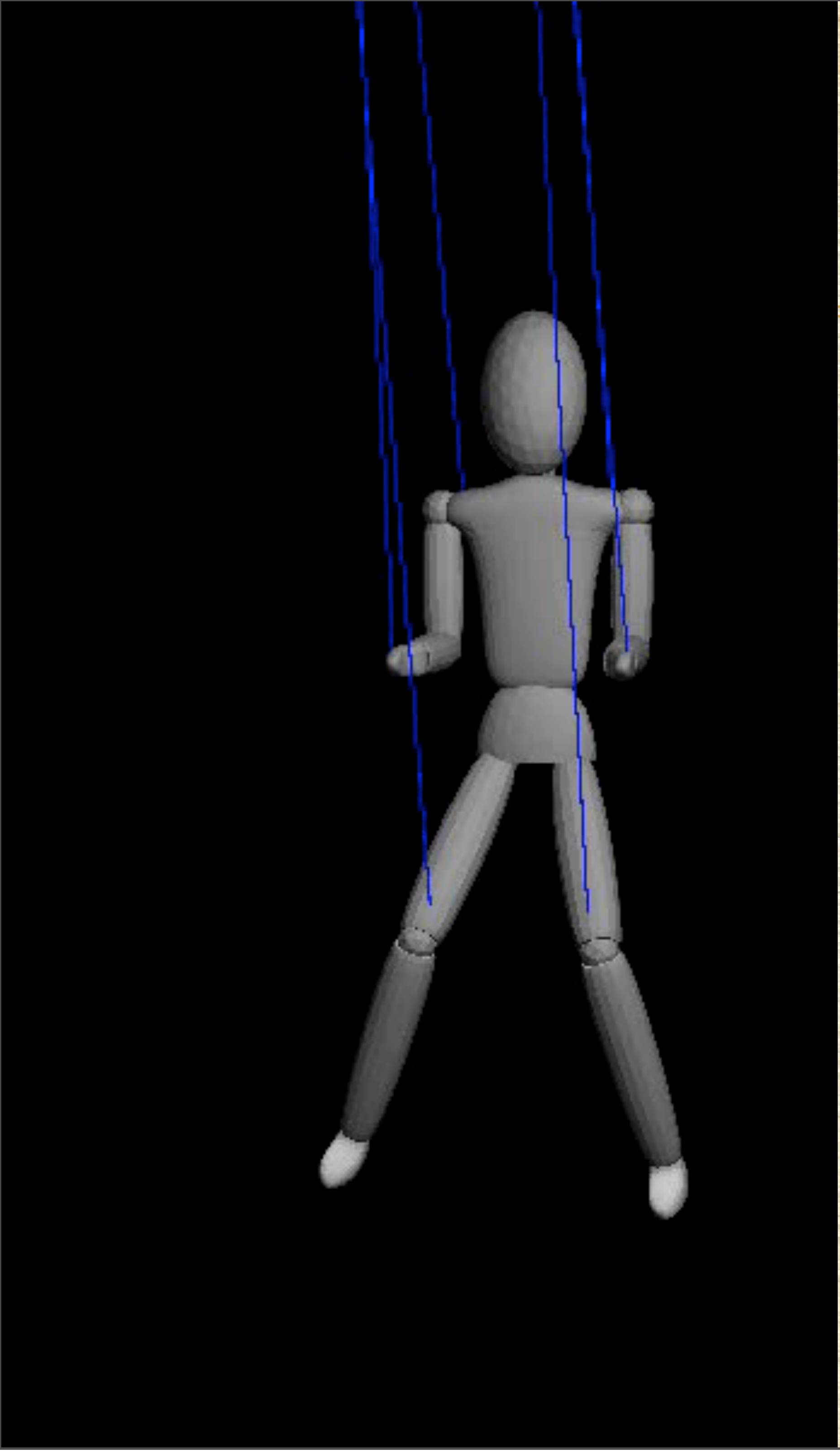
After 1000 seconds...



Variational integrator is a little out of phase

Computational Complexity

- Complexity primarily comes from inversion of the inertia tensor (just as in the continuous case)
  - special choice of coordinates can make this $O(n)$
- However, our implementation is somewhat worse than $O(n)$
- Nevertheless, it generally performs better for constrained systems because of superior convergence properties

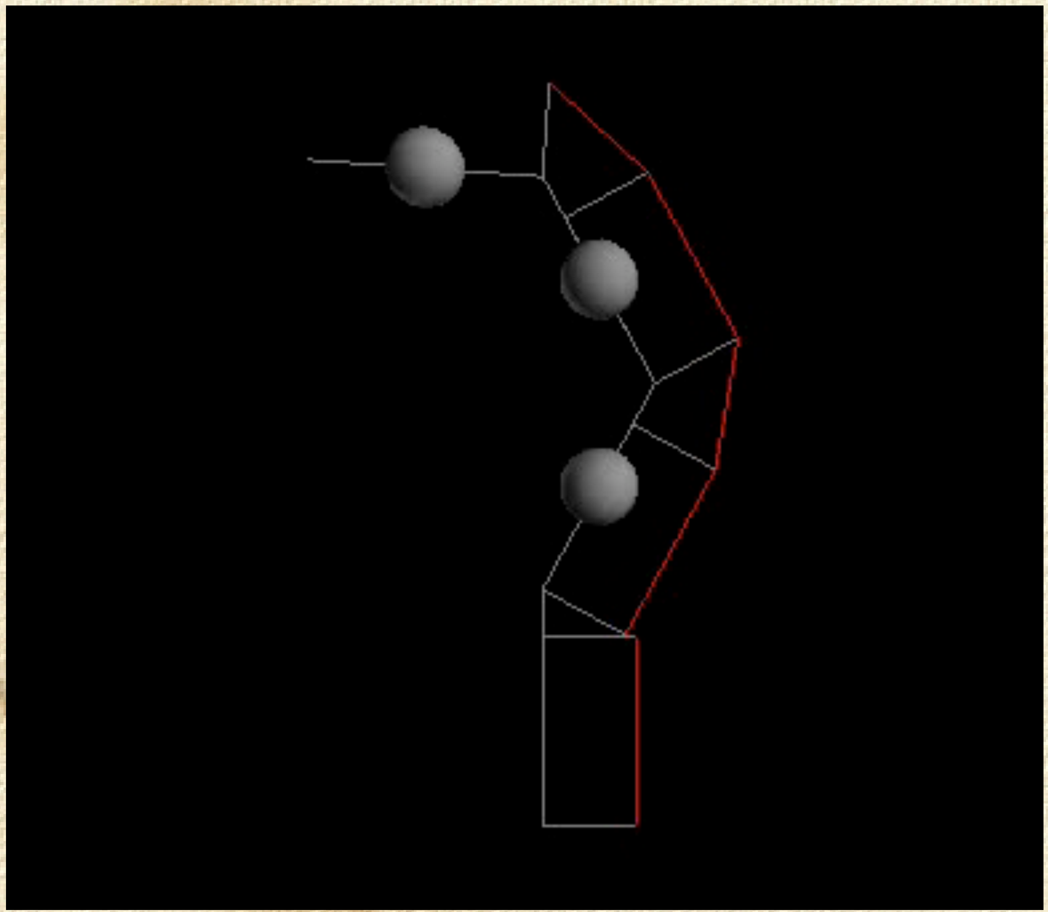


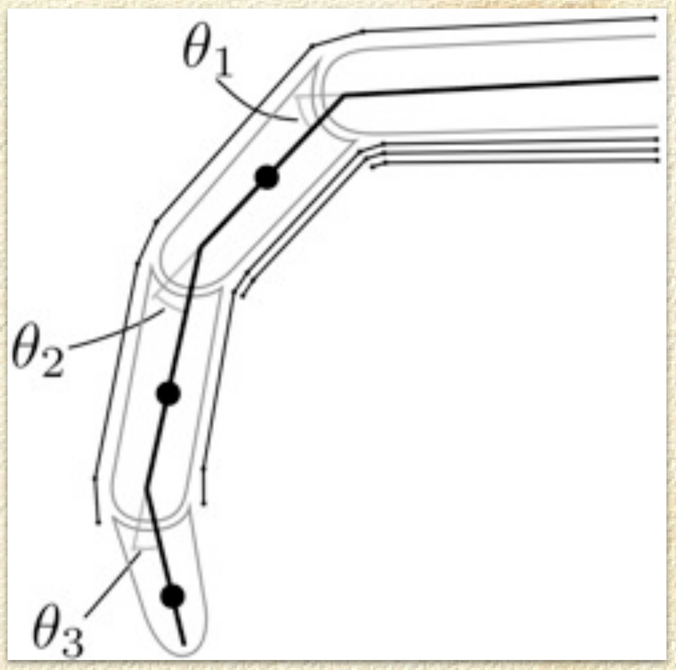
3D Kinematic Closed Chains

- Five simulations, using time steps of 0.01, 0.02, 0.03, 0.05, and 0.10
- The constraints are maintained
- The behavior is plausible even for 0.10 time steps!
- Computational time per iteration does not change much, so 0.10 is roughly 10x faster than 0.01.
- This example is what got us into hand dynamics

Beta available at
http://trep.sourceforge.net

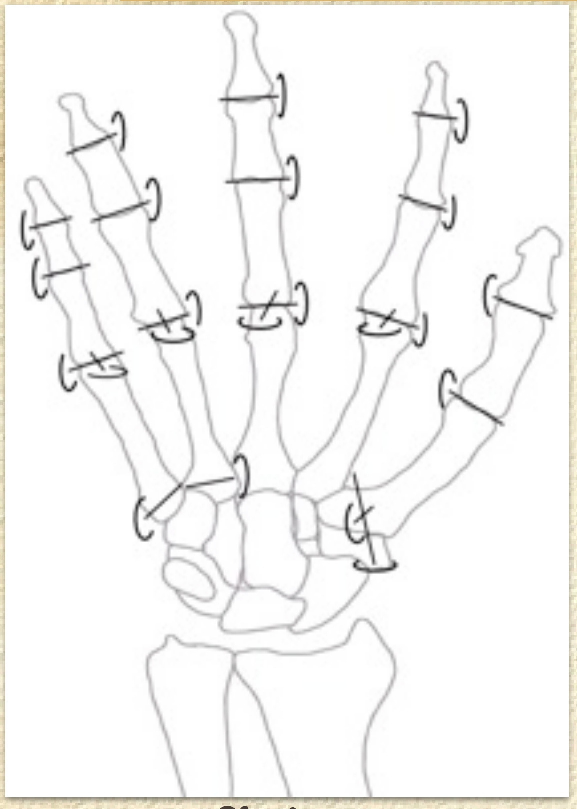
Finger Dynamics



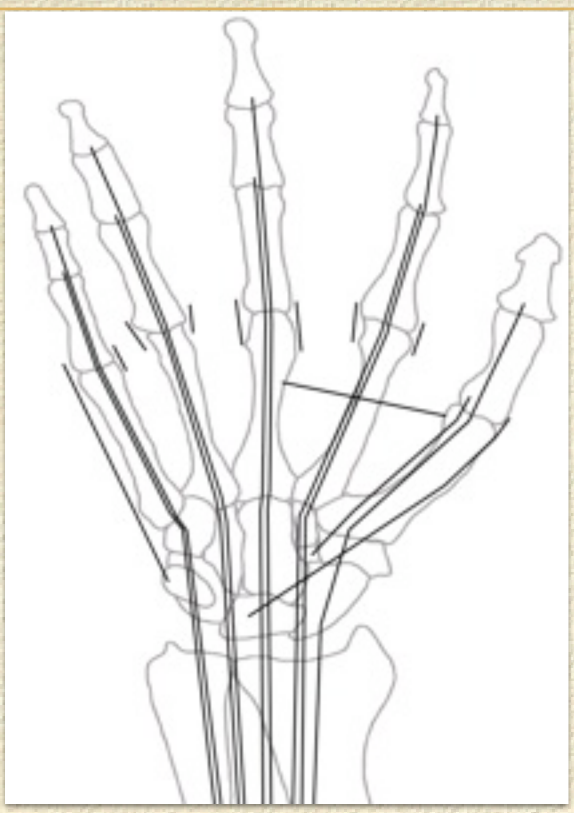


- Red line is strand
- Spheres represent masses and inertias
- Fully actuated, but strongly coupled

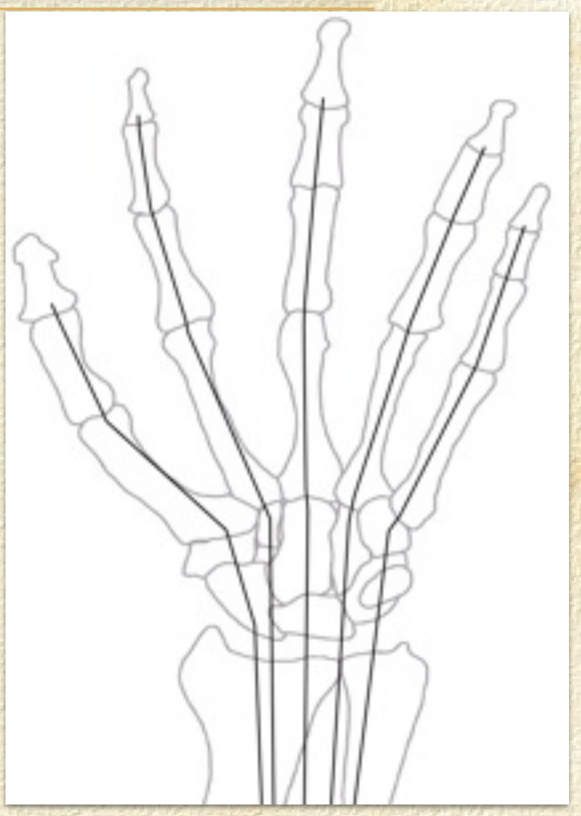
Strand-Based Mechanics



Joints



Palmar Strands



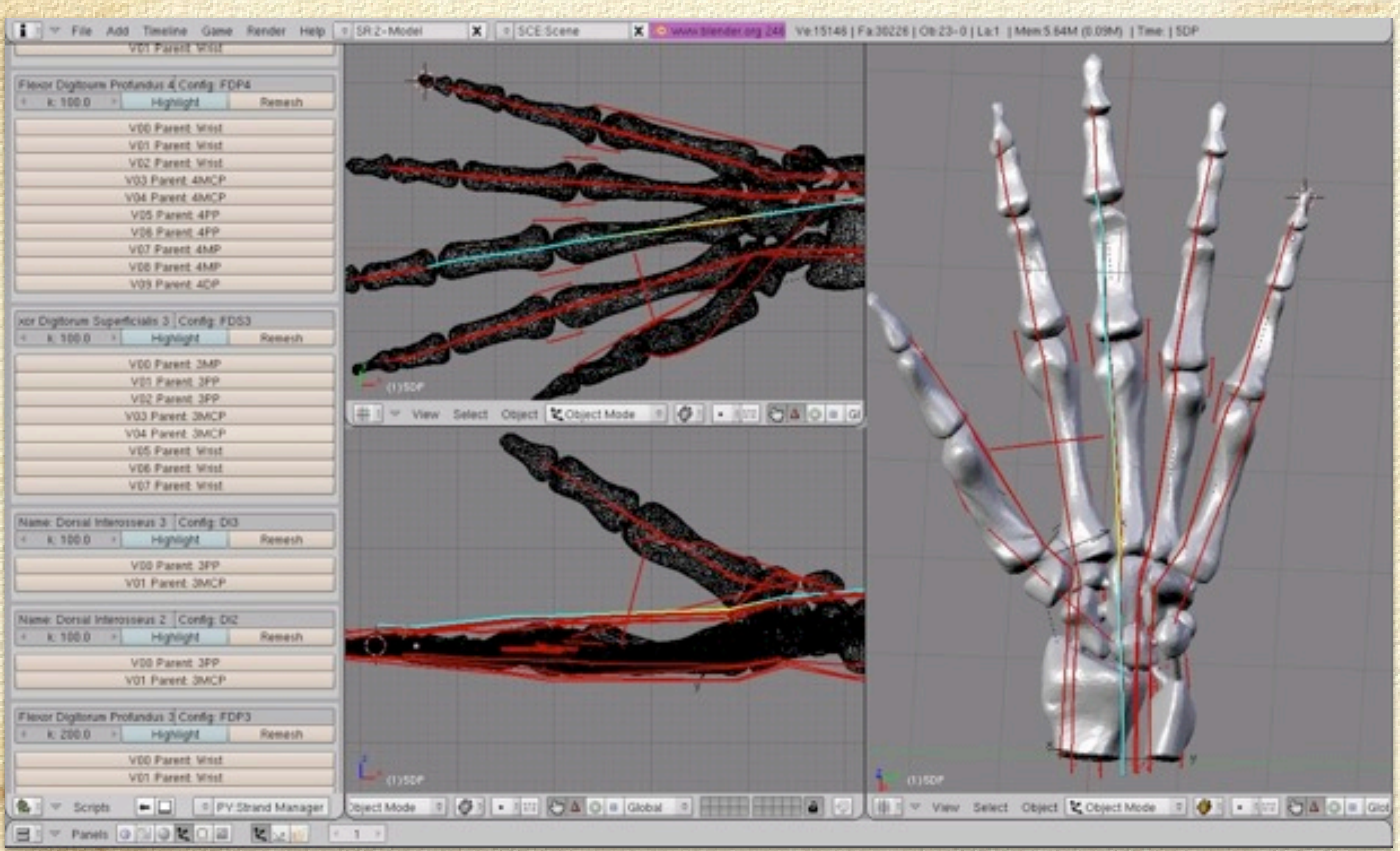
Dorsal Strands

- Joints and strands are chosen based on skeleton kinematics and hand dissections
- Includes significant coupling between strands
- Controllability is likely degenerate, but how would we know?

Unconstrained Hand

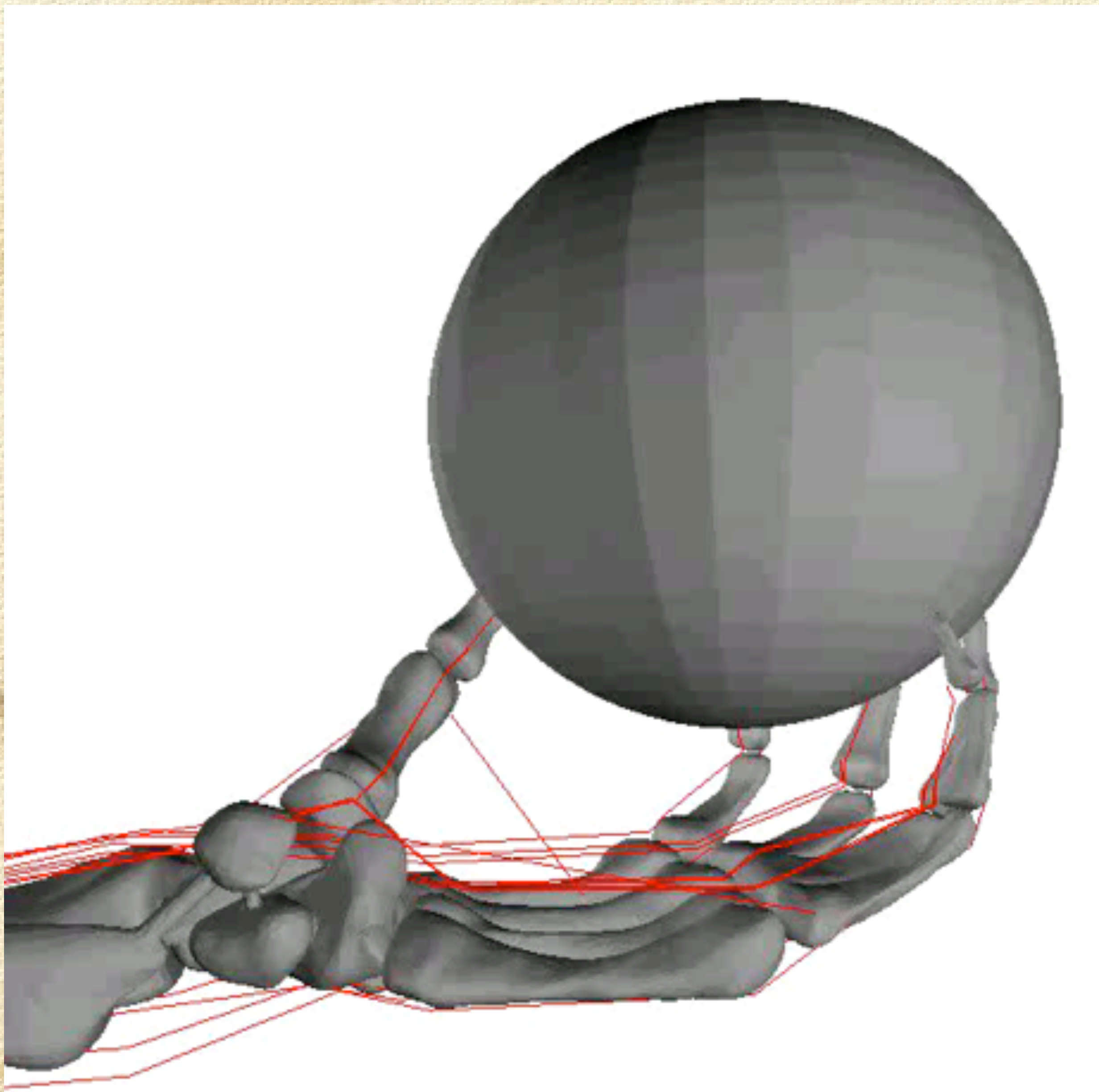
- Incorporates 19 rigid bodies and 23 muscle strands
- Strand model is a potential--no need to calculate the constraint forces
- Putting these models together requires expertise of experienced physiologist

Software



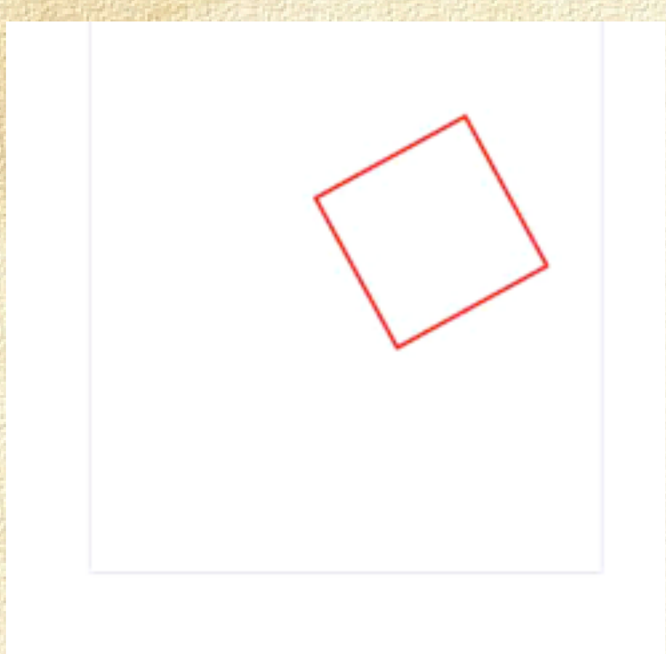
Points of attachment and material properties are not unique, thus requiring a *stable* prototyping environment

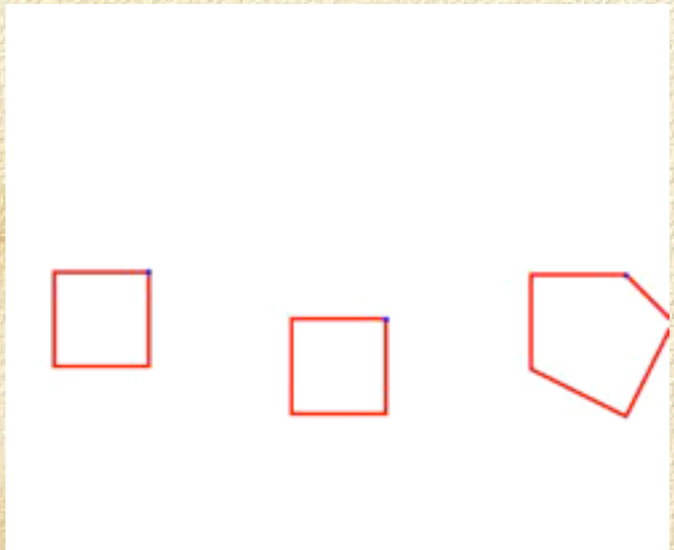
Constrained Hand



- hand holding sphere is still numerically stable with no parameter tuning
- again, this is essential to system identification
- because the simulation is not specific to hands, adding a sphere (or other shape) is easy

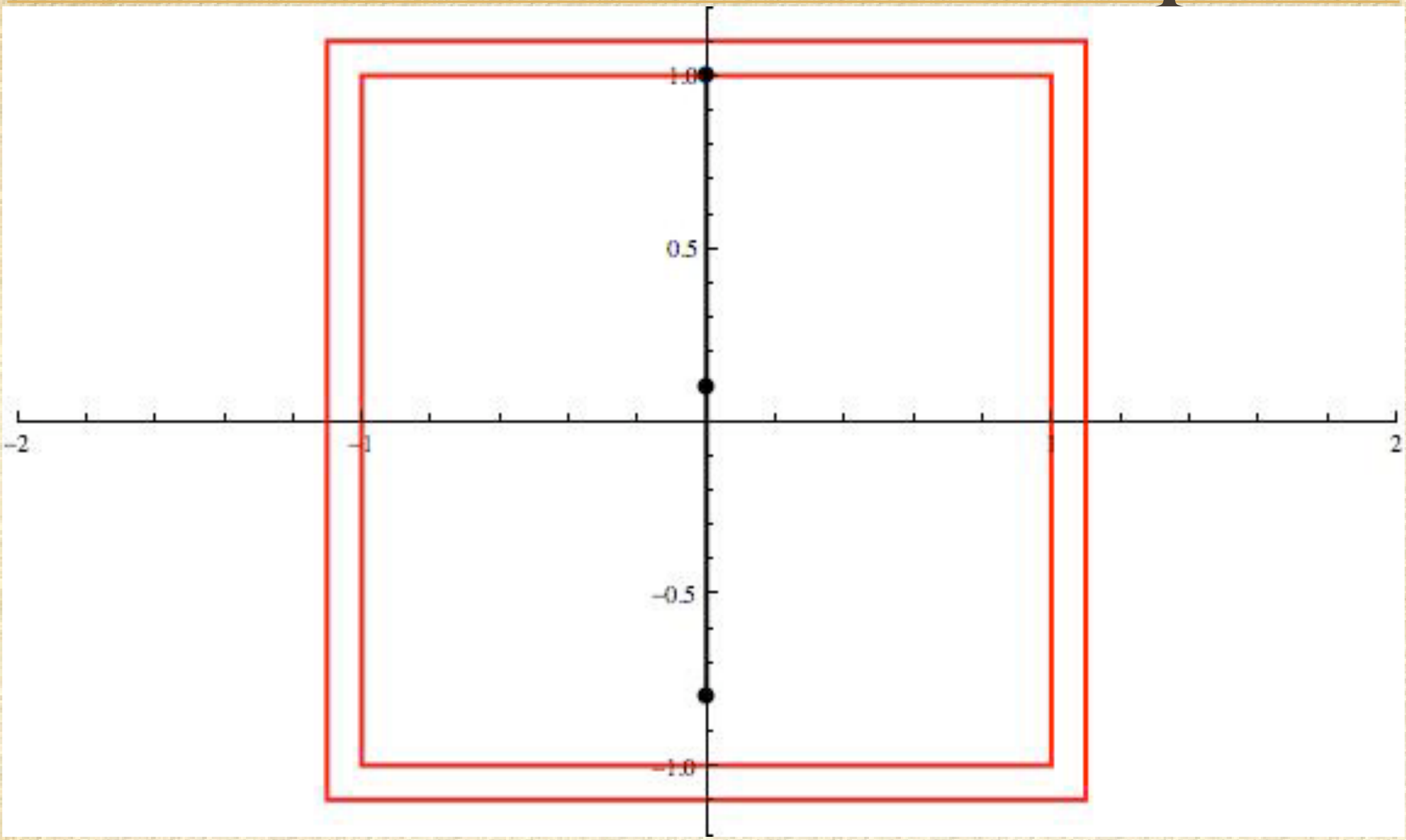
Worth Noting:
VIs also Good for Impacts





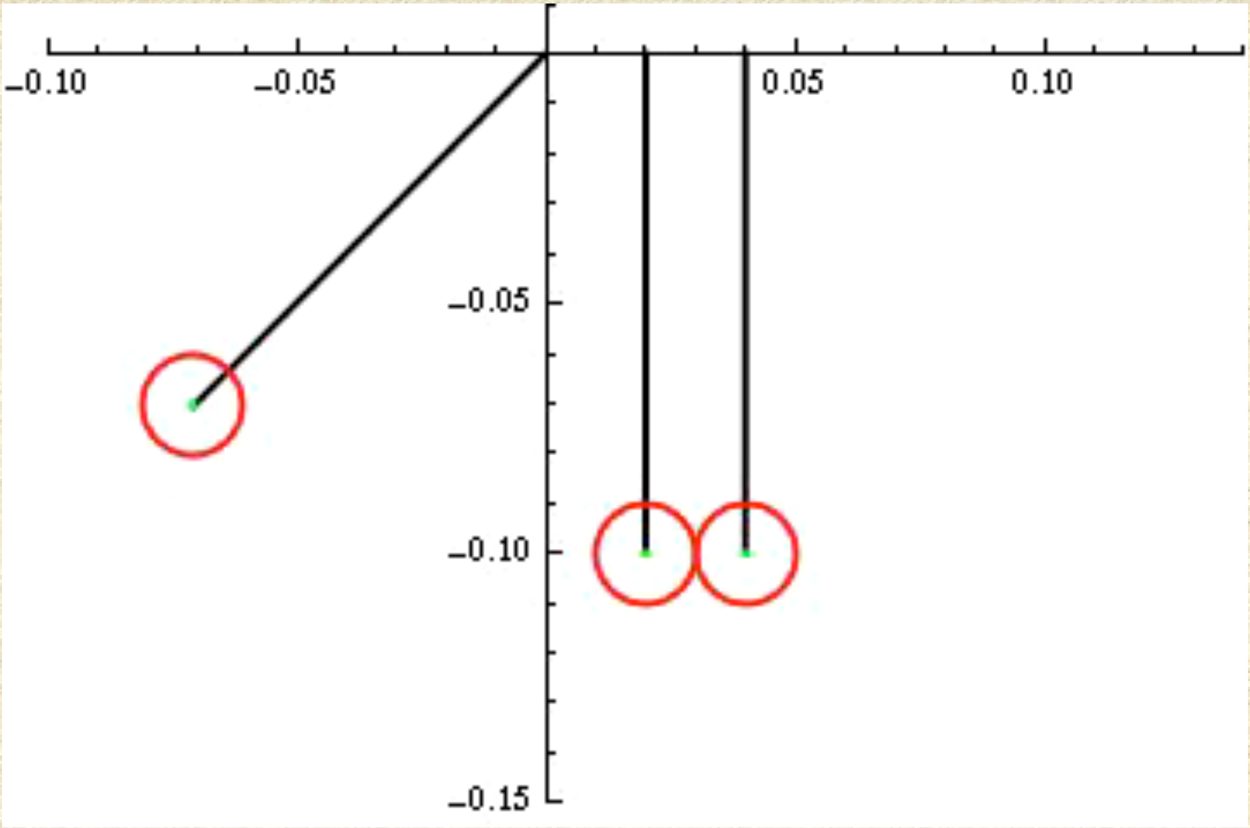
- Elastic and Inelastic impacts
- Completely avoids solving for impulses and complementarity problems
- Key to (eventually) simulating and identifying hand mechanics in grasping situations
- Can handle hundreds of impacts in a stable manner--see Fetecau et al. *SIAM J. Applied Dynamical Systems* Vol 2, No. 3, pp.381-416

Worth Noting: VIs also Good for Impacts



- A body that is not inertially fixed may hit itself, thereby creating a dynamic closed kinematic chain at impact (not implemented in *trep* yet)

Simultaneous Impacts
Newton's Cradle



- Newton's cradle experiences simultaneous impacts
- Still resolvable if choices of coordinates is made well
  - implies need for coordinate invariance

Current and Future Work: Limitations

- Why do variational integrators work so well in finite precision settings?
  - balance between integrals of motion and convergence
  - balance between dynamics and constraints
- Simulation by itself is not sufficient for solving most of the standing problems just discussed
  - System identification typically requires linearization
  - Contact mechanics require geometry
  - Kinematic and other purely geometric structures

Current and Future Work: Conclusions

- Variational integrators have the several advantages for hand simulation and other complex mechanical systems
  - very numerically stable
  - algorithmic specification
  - system identification is meaningful
  - contact mechanics are well-posed
- System identification with cadavers
  - work with Francisco Valero-Cuevas at USC
- Eventually we want to use this to optimize tendon networks for grasp functionality
- Motion Description Languages for marionettes

Current and Future Work: Conclusions

- Structural representations of systems is about more than simulation--it is also about analysis
  - linearization
  - optimization (e.g., system identification)
  - nonlinear controllability
  - reduction
- Example: Second-order hybrid optimization techniques that converge in ~10 iterations for problems that never converge using gradient descent
  - but they require second derivatives of continuous dynamics

Bibliography

- E. Johnson and T.D. Murphey, "Scalable variational integrators for constrained mechanical systems in generalized coordinates," IEEE Transactions on Robotics, Accepted
- E. Johnson, K. Morris, and T. D. Murphey, Algorithmic Foundations of Robotics VIII, ch. A Variational Approach to Strand-Based Modeling of the Human Hand. Springer-Verlag, 2008

Support and Thanks

CMS, ALT, CreativeIT,SRS

AFRL

Collaborators

Magnus Egerstedt — Georgia Tech

Francisco Valero-Cuevas — USC

Students

- Matthew Travers
- Elliot Johnson
- Karen Morris
- Corrina Gibson
- Tim Caldwell
- Matanya Horowitz
- Kirk Nichols