A Variational Approach to Strand-Based Modeling of the Human Hand

Elliot R. Johnson[2], Karen Morris[1], and Todd D. Murphey[2]

[1] Northwestern University Interdepartmental Neuroscience program, Northwestern University
[2] Department of Mechanical Engineering, Northwestern University

Abstract: This paper presents a numerical modeling technique for dynamically modeling a human hand. We use a strand-based method of modeling the muscles. Our technique represents a compromise between capturing the full dynamics of the tissue mechanics and the need for computationally efficient representations for control design and multiple simulations appropriate for statistical planning tools of the hand. We show how to derive a strand-based model in a variational integrator context. Variational integrators are particularly well-suited to resolving closed-kinematic chains, making them appropriate for hand modeling. We demonstrate the technique first with a detailed exposition of modeling an index finger, and then extend the model to a full hand with 19 rigid bodies and 23 muscle strands. We end with a discussion of future work, including the need for impact handling, surface friction representations, and system identification.

1 Introduction

Dynamic models of the hand are difficult to create because of strong coupling between every part of the system. Coupling arises from the geometry of the muscles and tendons (e.g. a single tendon affects multiple joints rather than just one), the physical structure (e.g. coupling between muscles), and neurological factors. Physiologists typically reduce the complexity by focusing on a single finger or joint. Roboticists often simplify the hand's complicated tendon geometry into models that apply torque directly at each joint. While appropriate for *engineered* systems (e.g. the RIC hand [1]–but not the Shadow Hand [9] which is cable-driven), this approach discards the coupling and geometry that are crucial for studying the mechanical capabilities and control strategies of the hand.

Studies of complete hands in physiology have mostly been restricted to static models because of complexity. Static models are useful for applications such as predicting fatigue and maximum finger-tip forces in equilibrium, but they are fundamentally limited. The hand activates different muscles and activation patterns in dynamic motions compared to static contractions, even

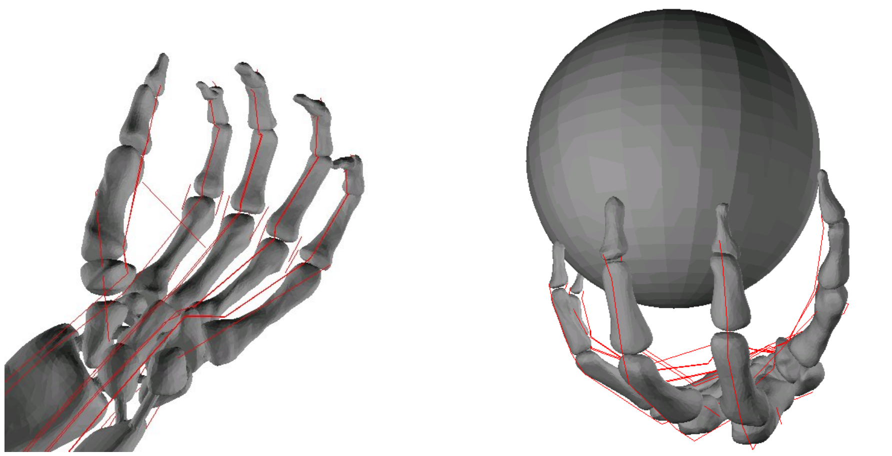


Fig. 1: The dynamic model of a human hand. (Left) The thin lines represent the muscle/tendon strands that actuate the hand. (Right) Contact can be modeled by holonomic constraints between finger tips and an object. (*The STL model was derived from http://www-static.cc.gatech.edu/projects/large_models/hand.html*)

when moving through identical postures [22]. Dynamic models are clearly an important yet underdeveloped research area.

This paper presents a dynamic model of a complete hand that is free of numerical dissipation. The muscle/tendon pairs are modeled as one degree of freedom (DOF) elastic elements (e.g. linear or nonlinear springs) called *strands* [2, 20]. The model, including muscle strands, is shown in Fig. 1. Modeling is accomplished with a tree-based [7] variational integrator [21]. The tree approach provides a consistent framework for describing a system's geometry and including elements like forces, constraints, and potential energies. Variational integrators have excellent numeric stability and energy conservation properties, and handle closed kinematic chains extremely well. They also behave well with both elastic and plastic impacts as well as friction. Together, variational integrators and tree-form representations provide stable, physically accurate simulations in generalized coordinates even for mechanically complex systems like the hand.

Variational integrators should be of particular interest to the computer science part of the robotics community. They represent dynamics in a naturally discrete setting–hence the term *discrete mechanics*–rather than a continuous one. This feature, along with the fact that variational integrators are valid over long time horizons, makes probabilistic planning and optimal control more feasible. Moreover, because variational integrators are particularly simple to implement they can be integrated easily into other algorithms. Lastly, algorithms capable of computing variational integrators typically can compute other quantities such as the linearization; this allows one to evaluate the local singular value decomposition of a system at any operating point. The key point is that variational integrators provide a particularly computational view of mechanics, starting from their derivation, and this is useful when working with complex systems in a robotics context.

This work demonstrates that a variational modeling approach leads to numerically stable models that are easy to modify and extend. The model includes dynamics without resorting to PDEs, favoring useful abstractions and computational tractability instead. Variational integrators also deal with constraints (particularly closed kinematic chains) in a natural and robust way. Hence, these techniques can also be used on simpler systems such as the Shadow Hand [9] and other cable-driven robots.

We begin with a background discussion of hand anatomy and discuss previously published models in Sec. 2. We continue with an overview of variational integrators and discrete mechanics in Sec. 3. Section 4 discusses the strand abstraction our model uses for muscles/tendons in the hand. Finally, we present several simulations in Sec. 5 that use our model.

2 Background: Hand Modeling

Hand physiology is a complex subject that involves understanding the mechanical structure (i.e. system identification), characterizing the nervous system's control strategies, and developing mechanical models of the hand. We present a brief overview of hand anatomy and modeling strategies.

2.1 Hand Anatomy

The human hand contains 27 bones [6] with approximately 30 degrees of freedom [12]. It is actuated by 29 muscles, some of which are subdivided into parts that contract independently to provide a total of 38 unique actuators. Modeling the hand involves complex geometry due to the joints of the skeleton, sliding of the muscles, and the routing of the tendons.

We focus on the hand from the wrist to the finger tips. Each of the four digits is associated with four bones: the metacarpal (MCP), proximal phalanx (PP), middle phalanx (MP), and distal phalanx (DP). There are approximately five DOF for each digit: one at the base of the MCP, two at the MCP/PP joint, one at the PP/MP joint, and one at MP/DP joint.

The thumb has three bones (MCP, PP, and DP) and approximately four DOF allowing it to oppose, abduct, adduct, and flex [5]. The muscles actuating the thumb have complex geometry compared to those for the digits. As a result, the thumb is often not modeled despite contributing 40% of the function of the hand [8]. The abstractions used in this paper scale well with this type of complexity, and so the thumb *is* included in our model.

Bone geometry plays a significant role in hand dynamics. Tendons slide along bone surfaces, changing where forces are applied as the bones move. Because of the role that bone geometry plays, rapid prototyping is essential

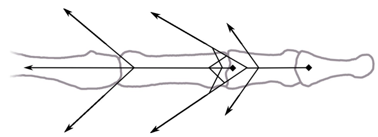


Fig. 2: The dark lines are a schematic representation of an extensor tendon. The single tendon attaches to several bones and muscles. Diamonds indicate fixed insertions to bones. Arrows represent muscle forces. The complex geometry results in strong coupling between muscles and non-trivial control.

for generating meaningful hand models; physiologists often have insight about how a muscle/tendon moves across a bone during hand motion.

One of the hardest aspects of hand models are the tendons. Tendons are made up of dense connective tissue that is elastic, flexible, and strong [18, 5]. We will divide types of tendon into three classes for our purposes.

The first class we will define is made of simple tendons that are short connectors between a muscle and a single bone. The tendons that connect muscles to their origins are typically this simple type.

The second class has slightly more complicated tendons that connect one muscle to two bones, but work over long pathways. These tendons may slide over many bones, joints and fibrous sheaths that act as pulleys. As a result, the tension in the tendon applies forces to multiple points along its path and creates coupling between joints [19]. The flexor tendons of the hand are examples of this class.

Finally, the most complex tendons connect multiple bones and muscles, and also work over long pathways. These tendons can branch off and connect in many places to form complex structures. The extensor tendons, shown in Fig. 2 are this type. Identifying accurate representations of these tendons is still an open research problem in physiology [23].

Dynamic modeling of the hand is complicated by other physiological factors as well. The hand is over-actuated [6] and kinematically redundant [4] so there is no one-to-one mapping between muscles and joint torques. There is also significant coupling between muscles caused by both mechanical coupling and activation of more than one muscle at a time. This is again distinct from traditional robotic hands where all joints are independently controlled. Coupling plays an important role in all aspects of hand motion. Muscles are often co-activated such that some provide the major force while others stabilize the motion.

2.2 Hand Modeling

Partial differential equations have been used for modeling walking [15], facial movements [17], the surface of the heart [10], etc. However, there is no intrinsic reason to believe that a model that simplifies the hand system to a series of one degree-of-freedom-spring kinematic chains would be any less accurate than a highly complex PDE simulation [15]. This is because PDE models necessarily make assumptions about the underlying homogeneity of the muscle and bone tissue, leading to models that cannot be identified well. Moreover, for control purposes we wish to have a “simple” model that incorporates all the hand geometry and coupling in a dynamically correct manner.

Assuming one accepts a finite dimensional modeling setting, a finger can be modeled as a kinematic chain [13]. Lee and Kroemer [11] created a kinematic model of the finger that included flexion and extension. The moment arms of the tendons are constant and external forces were considered. This model was used to measure finger strength. Static equilibrium problems were used to make inferences about the dynamics of the model. Notably, there has been relatively little work on whole hand modeling.

Muscle models in the hand are often modeled as weightless expandable threads [2]. Models in the past that use the weightless expandable threads in hand modeling do so by solving static problems at each step and then animate the steps to create a smooth motions of the hand (see Sueda et al. [20]). The use of static poses and animation is effective at producing human like movement, but may not be natural. The model presented in this paper also uses the weightless expandable thread technique to model the tendons, but has the advantage of having no numerical dissipation (the major drawback described in [20]). Lastly, kinematic redundancy can be dealt with by adding constraints that reflect the physiology of the hand. These constraints can be easily included in our numerical simulation because variational integrators (described in the next section) are particularly well-suited to modeling closed kinematic chains.

3 Background: Variational Integrators

Variational integrators are a result of relatively recent research in discrete mechanics. These integrators are derived in a similar way as the Euler-Lagrange equations. They have been shown to respect important mechanical properties like conservation of energy and momentum (in the absence of nonconservative forcing), and have been observed to have other desirable properties like good dissipation modeling for systems with friction, excellent closed-kinematic chain behavior, and good convergence. Variational integrators also work directly in generalized coordinates which are preferred for describing anatomical aspects of the hand.

We introduce variational integrators with an overview of how the Euler-Lagrange equation is derived from a variational principle and discuss how the derivation is modified to obtain variational integrators.

Lagrangian mechanics provide a coordinate-invariant method for generating a system's dynamic equations. Lagrangian mechanics are derived from a variational principle. We define the Lagrangian of system as its total kinetic energy minus potential energy in terms of generalized coordinates $q = [q_1, q_2, \ldots, q_n] \in \mathbb{R}^n$ and their time derivative $\dot{q} = \left[\frac{\partial q_1}{\partial t}, \frac{\partial q_2}{\partial t}, \ldots, \frac{\partial q_n}{\partial t}\right]$:

$$L\left(q, \dot{q}\right) = KE(q, \dot{q}) - PE(q) \tag{1}$$

The integral of the Lagrangian over a trajectory is called the Action (Fig. 3a):

$$S(q([t_0, t_f])) = \int_{t_0}^{t_f} L(q(\tau), \dot{q}(\tau)) \mathrm{d}\tau \tag{2}$$

Hamilton's Least Action Principle states that a mechanical system will naturally follow the trajectory that extremizes (e.g. minimizes) the action; hence, it is not so much a least action principle as an action stationarity principle. Extremizing (2) with a variational principle shows that such trajectories satisfy the Euler-Lagrange equation:

$$\frac{\partial}{\partial t} \frac{\partial L}{\partial \dot{q}}(q, \dot{q}) - \frac{\partial L}{\partial q}(q, \dot{q}) = 0 \tag{3}$$

which is a second-order ordinary differential equation (ODE) in q. Given a set of initial condition $(q(t_0), \dot{q}(t_0))$, we numerically integrate (3) to simulate the system dynamics. The derivation can be extended to include holonomic/non-holonomic constraints, external forces, and dissipation [14].

In derivation of (3), the system's trajectory is always continuous. The trajectory is not discretized until the last step during numeric integration. A variational integrator, on the other hand, is derived by introducing the time discretization before applying the variational principle.

3.1 Discrete Mechanics

In discrete mechanics, we seek a sequence $\{(t_0, q_0), (t_1, q_1), \ldots, (t_n, q_n)\}$ that approximates the actual trajectory of a mechanical system ($q_k \approx q(t_k)$). In this paper, we assume a constant time-step ($t_{k+1} - t_k = \Delta t \ \forall \ k$) for simplicity, but in general, the time-step can be varied to use adaptive time-stepping algorithms.

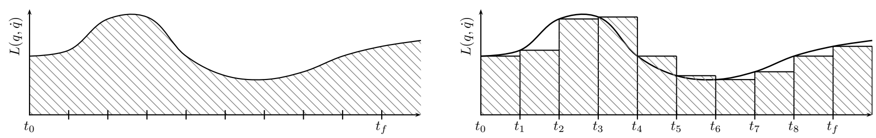
A variational integrator is derived by defining a discrete Lagrangian, L_d, that approximates the continuous action integral over a short time interval.

$$L_d\left(q_k, q_{k+1}\right) \approx \int_{t_k}^{t_{k+1}} L(q(\tau), \dot{q}(\tau)) \mathrm{d}\tau \tag{4}$$

The discrete Lagrangian allows us to replace the system's action integral with an approximating action sum.

$$S(q([t_0, t_f])) = \int_{t_0}^{t_f} L(q(\tau), \dot{q}(\tau)) \mathrm{d}\tau \approx \sum_{k=0}^{n-1} L_d\left(q_k, q_{k+1}\right) \tag{5}$$

where $t_f = t_n$. This approximation is illustrated in Fig. 3. The shaded region in Fig. 3a represents the continuous action integral. The shaded boxes in Fig. 3b represent values of the discrete Lagrangian, which are summed to calculate the discrete action.



(a) Continuous Action Integral (b) Discrete Action Sum

Fig. 3: The discrete action sum approximates the continuous action integral using the discrete Lagrangian (which requires a choice of quadrature rule, in this case the standard Riemann integral).

In continuous mechanics, a variational principle is applied to extremize the action integral and derive the well-known Euler-Lagrange equation. The same approach is used to extremize (5) to get the discrete Euler-Lagrange (DEL) equation[1].

$$D_1 L_d \left(q_k, q_{k+1}\right) + D_2 L_d \left(q_{k-1}, q_k\right) = 0 \tag{6}$$

$$h(q) = 0 \tag{7}$$

where Eq. (7)represents constraints if there are any. Nonholonomic constraints can be represented as well, but for simplicity we do not discuss them here.

This is an implicit difference equation that depends on the previous, current, and future states. Given q_{k-1} and q_k, (6) is treated as a *root-finding* problem to find q_{k+1}. After advancing k, this process is repeated to simulate the system for as long as desired.

Constraints play a crucial role in simulations with contact and grasping. We can often represent mechanical contact with a holonomic constraint (i.e. a constraint on the system's configuration manifold). In continuous mechanics, holonomic constraints are typically differentiated and included as velocity/acceleration constraints. Over time, numeric integration errors build up and the trajectory violates the original holonomic constraint. In differential-algebraic techniques that enforce the constraint, the error is either accepted or corrected with heuristic methods that (as a side effect) artificially inject or dissipate energy. This is acceptable for computer entertainment applications, but leads to unrealistic system identification and unstable controllers for physical applications.

Variational integrators avoid this problem and implement holonomic constraints well as a result. At each time step, the constraints $h(q)$ are satisfied

[1] $D_n f(\ldots)$ is the derivative of $f(\ldots)$ with respect to its n-th argument. This is sometimes called the *slot derivative*

(to within the tolerance of the numeric root solver) by appending them to (6) and including a constraint forcing term in the discrete Euler-Lagrange equation. The constraint resolution is also directly coupled to the dynamics via the discrete Lagrange D'Alembert principle rather than being a heuristic fix as in the continuous case.

3.2 Complexity

A common problem with Euler-Lagrange simulations is that the equations grow too quickly as the system becomes larger (or more complex). Most simulation methods are therefore based on force balance methods and avoid generalized coordinates. Our recent work [7] discusses a new approach to Lagrangian simulations that significantly reduces the complexity growth and keeps generalized coordinates feasible for much larger systems. This approach ensures through the use of caching in a tree-structure that every transformation is only computed once, trivially leading to $O(n)$ computation of Eq. (6) for unconstrained systems (for constrained system one gets $O(n + m)$ to compute Eqs. (6) and (7) with m constraints). However, most root-solving techniques require differentiating Eqs. (6) and (7) so that Newton's Method can be applied. This entails inverting an nn matrix, which in many cases is also an $O(n)$ calculation so long as one is careful to take advantage of the group structure.

It is worth noting here that the computational complexity of computing $f(\cdot)$ in $\dot{x} = f(x, u)$ is not always the relevant notion of complexity; it assumes that we are only interested in the complexity of evaluating a single step of an integrator. Rather, we are typically interested in knowing the complexity of obtaining a solution that is within some error of the "true" solution to the equations of motion. We have an example in [7] of a system–the scissor lift–that scales linearly in terms of computing $f(\cdot)$ but scales *exponentially* in terms of computing the correct solution. This point should not be taken lightly–it implies that one of the main metrics we use for evaluating simulation techniques is often times off-point. For that example, variational integrators are substantially more efficient at computing the correct solution even though they are less efficient at computing Eqs. (6) and (7).

We have implemented these ideas in a freely available, open-source package called `trep`[2]. `trep` is designed as an easy to use tool for rapid, incremental development of simulations without sacrificing performance. It provides useful facilities like compact representations of systems and automatic visualization. Most importantly, `trep` is easy to extend with new types of potentials, forces, and constraints. This allows us, for example, to quickly explore different muscle/tendon representations in a common, structured environment. The ability to quickly adapt a simulation is critical to a low-dimensional model's success.

---

[2] http://trep.sourceforge.net

4 Strand Models of the Hand

We now move on to the use of strand models in a variational context. As previously mentioned, a simplified but useful model in physiology is to consider muscle/tendon groups as linear springs. A spring constant, k, is chosen to reflect the bulk elasticity of the the tendon and muscle tissue. The muscle is contracted and relaxed by controlling the natural length, x_0, of the spring.

For the spring model to be useful for hand models, we must extend it to handle the routing and sliding needed for extrinsic muscle tendons. We can think of a muscle/tendon as a *strand* that connects two points and slides through intermediate points.

Formally, a strand is defined by a spring constant, $k \in \mathbb{R}$, a (possibly time-varying) natural length $x_0 \in \mathbb{R}$, and a set of points $p_1, p_2 \dots p_N \in \mathbb{R}^3$ where $N \geq 2$. The current length of the strand is found by accumulating the linear distance between adjacent points:

$$\begin{aligned} x(q) &= \sum_{i=1}^{N-1} ||p_i(q) - p_{i+1}(q)|| \qquad (8) \\ &= \sum_{i=1}^{N-1} \left[(p_i(q) - p_{i+1}(q))^T (p_i(q) - p_{i+1}(q)) \right]^{\frac{1}{2}} \end{aligned}$$

The potential energy of the strand is then:

$$V(q,t) = \tfrac{1}{2} k \left(x(q) - x_0(t) \right)^2 . \qquad (9)$$

As one would expect, one can replace a nonlinear potential with the quadratic one. `trep` also requires the derivative of (9) to implement the strand potential. These are straightforward to calculate by applying chain rule.

$$\frac{\partial V}{\partial q}(q,t) = k \left(x(q) - x_0(t) \right) \cdot \frac{\partial x}{\partial q}(q)$$

The derivative of $x(q)$ is found similarly.

$$\begin{aligned} \frac{\partial x}{\partial q_j}(q) &= \sum_{i=1}^{N-1} \Bigg(\tfrac{1}{2} \left[(p_i(q) - p_{i+1}(q))^T (p_i(q) - p_{i+1}(q)) \right]^{-\frac{1}{2}} \cdot \\ &\qquad \Bigg[\left(\frac{\partial p_i}{\partial q_j}(q) - \frac{\partial p_{i+1}}{\partial q_j}(q) \right)^T (p_i(q) - p_{i+1}(q)) + \\ &\qquad (p_i(q) - p_{i+1}(q))^T \left(\frac{\partial p_i}{\partial q_j}(q) - \frac{\partial p_{i+1}}{\partial q_j}(q) \right) \Bigg] \Bigg) \\ &= \sum_{i=1}^{N-1} \frac{(p_i(q) - p_{i+1}(q))^T \left(\frac{\partial p_i}{\partial q_j}(q) - \frac{\partial p_{i+1}}{\partial q_j}(q) \right)}{||p_i(q) - p_{i+1}(q)||} \end{aligned}$$

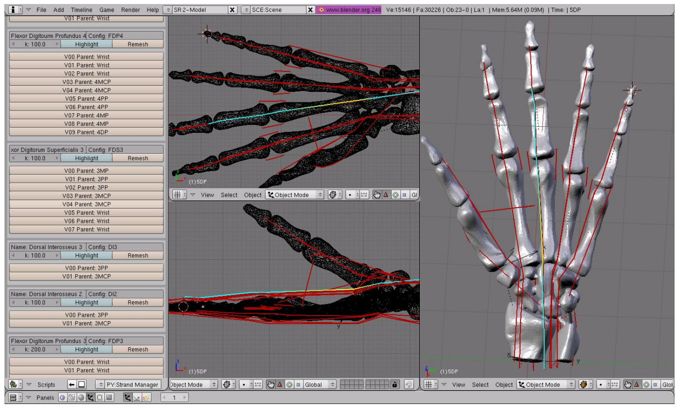


Fig. 4: A graphical user interface (GUI) makes it easy to integrate new muscle strands in the model.

There are a number of ways this model can be improved. The most simple improvement is to use non-linear potentials instead of a linear model. Once a better potential shape is identified experimentally, it can be used by updating (9) and (4). The strands should also be extended to include branching and sliding so that complex tendons like the digit extensors can be modeled correctly. However, this will require system identification to determine how the topology should be defined; real tendons do not join at unique locations and are instead defined by large, somewhat amorphous regions of connection.

5 Model Implementation

The strand model was implemented as a new potential type in `trep`. We created a 2D finger model (Sec. 5.1) and a full 3D hand model (Sec. 5.2). A third model is presented by adding a sphere and holonomic constraints to simulate grasping contact. Hand dimensions were based on the STL model in Fig. 1 and the spring constants were chosen to be rather stiff (100-300 N/m) to represent the stiffness of the tendons. However, careful system identification using the whole model should be done in the future (along the lines of [16]) to properly calculate the model parameters.

Simulations were developed using Blender[3] as a graphical user interface (GUI). Figure 4 shows a custom plug-in provides convenient ways to define and modify new strands. Blender was also used to define the desired trajectories and poses. The combination of `trep` and Blender makes extending and improving the model easy.

The sources for the hand model can be downloaded from `trep`'s website at http://trep.sourceforge.net/examples/hand.html.

[3] http://www.blender.org

5.1 A Finger

A two-dimensional finger is simple compared to a full hand, but also easier to understand. The model is shown in Fig. 5. We have simplified the extensor tendon into a simple tendon with one muscle. The model has three degrees of freedom and four muscle strands.

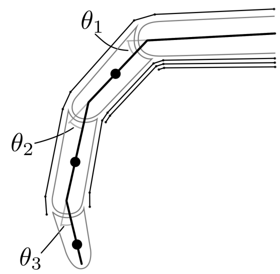


Fig. 5: The 2D finger model has three degrees of freedom and four muscles/tendon strands.

The simulation was created by defining a desired trajectory and computing the corresponding natural lengths for each strand. The dynamic system was then simulated using these lengths as inputs. The result is a rudimentary control scheme that does not incorporate feedback. It is intended to demonstrate the dynamic model rather than accurately simulate hand control.

The model was simulated in `trep` for 30.0s with a time step of 0.01s. The simulation took approximately 34s to compute on a 2.2GHz Intel Core2 Duo processor. The results are shown in Fig. 6.

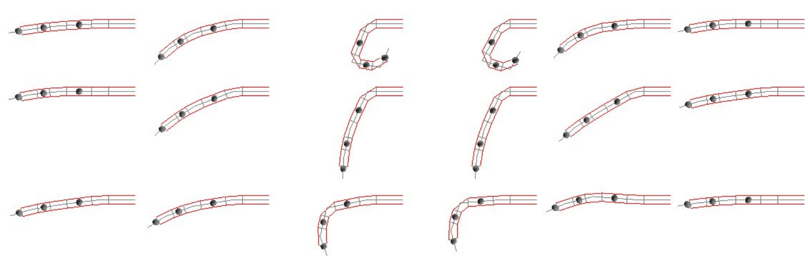


Fig. 6: The finger was moved through several motions. The strand model was capable of actuating the finger to follow the trajectory.

The trajectories for the three joint angles are plotted in Fig. 7. For each of the three motions, the joint angles tend to move in the same direction as a result of the *coupling* introduced by the tendons.

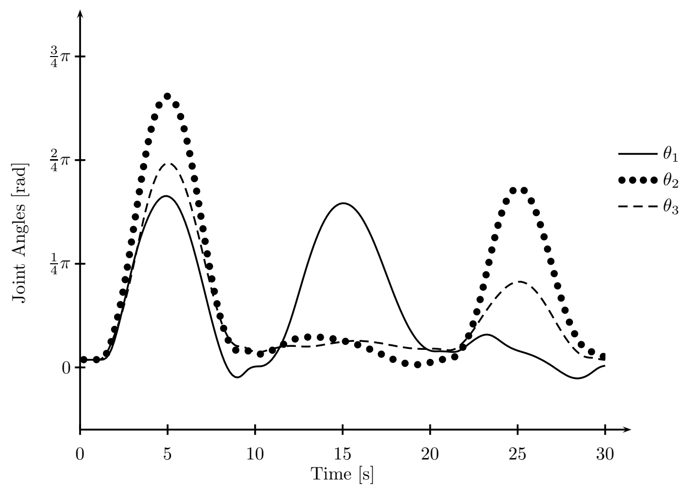


Fig. 7: Joint angles vs. time for a finger simulation. Note how all three angles tend to move in the same direction whenever there is movement because of coupling between joint angles.

5.2 Full Hand

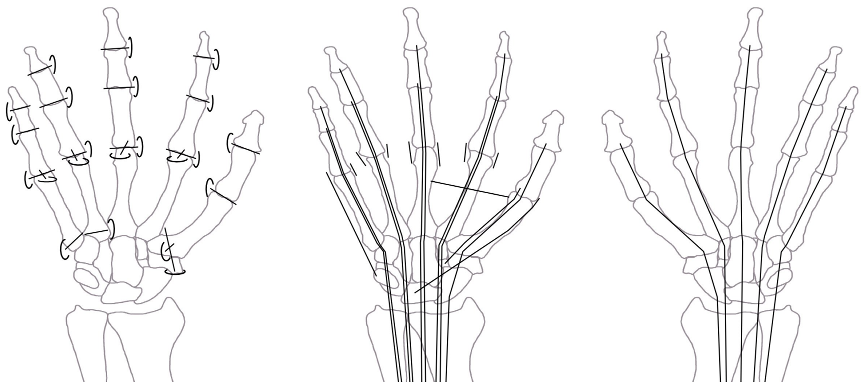
We extended the finger to a full hand model. The goal for this model is to demonstrate that the fingers can be individually actuated when there is no coupling between muscle groups; coupling and co-activation can always be added to this model without difficulty because once the tendon topology is known these simply add stress/strain relationships. Our model has 20 degrees of freedom along with 23 independently actuated strands. Fig. 8 illustrates the model.

The most significant simplification is that the extensor tendons have been divided into several different muscle/tendons.

The model was tested using a hand closure trajectory. The hand begins with all digits extended. The digits are flexed inwards toward the palm and then return to their original extended configurations. The simulation is defined and run in the same manner as the above finger simulation.

The simulation lasts for 10 seconds and uses a time step of 0.001s. A 2.2GHz Intel Core2 Duo Processor completed the simulation in approximately one hour. Several stages of the simulation are shown in Fig. 9.

The simulation is slow (compared to real-time) because of the small time steps that are required by large spring constants. However, the corresponding high-frequency oscillations are almost completely absent in the trajectory. This suggests that a stiff elastic tendon model may be inappropriate for grasping/large-movement simulations. A better model might take mus-



(a) Joints (b) Palmar Strands (c) Dorsal Strands

Fig. 8: The complete hand model including coupling between fingers.

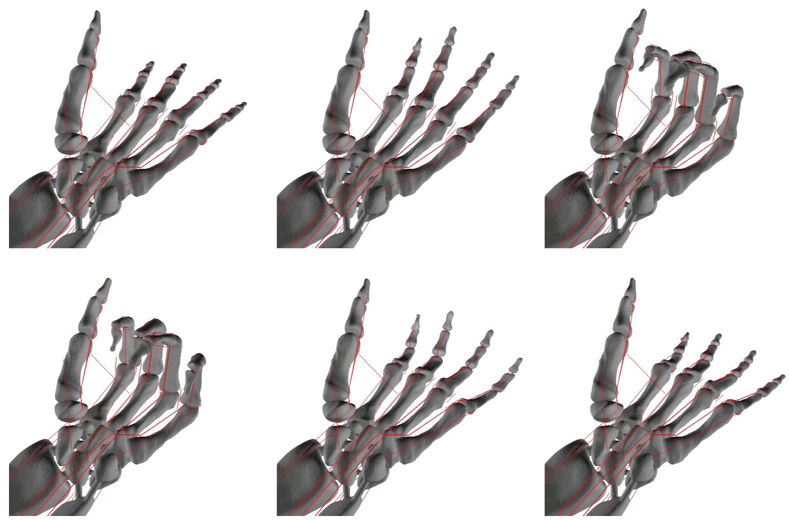


Fig. 9: This figure shows several stages of the simulation results. (Time advances left to right, top to bottom)

cle tension as input and use the strand geometry to determine the resulting forces on the hand. While this has been deferred for future work, the change is straightforward in `trep`.

5.3 Grasping Simulation

Finally, we demonstrate the extendibility of the model with a grasping simulation. A sphere was added and brought into contact with the five finger tips.

Holonomic constraints attach the finger tips to the sphere's surface, so we are modeling assuming "infinite" friction between the finger tip surface and the sphere surface. For this simulation, the natural length of each strand was fixed and gravity was added. The resulting trajectory is due entirely to the dynamic interaction between the sphere and hand.

The simulation ran for 10 seconds with a time step of 0.001s. The simulation was completed in approximately one and half hours on a 2.2GHz Intel Core2 Duo processor. Several stages of the simulation are shown in Fig. 10.

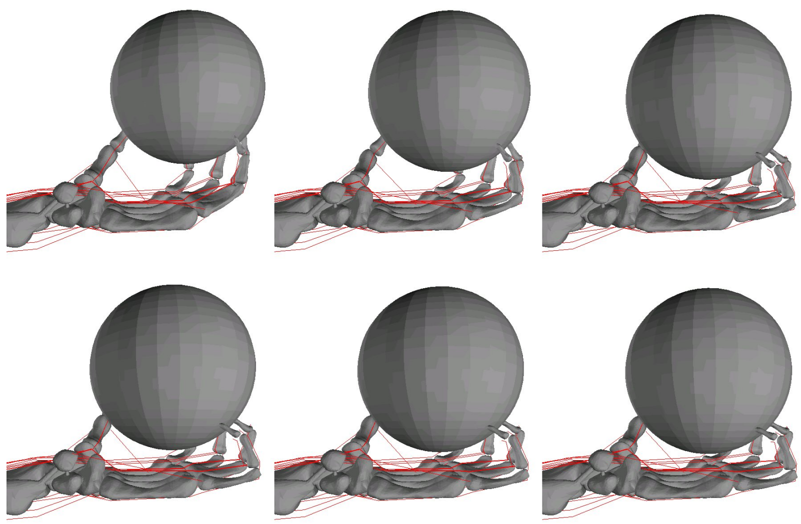


Fig. 10: This figure shows several stages of the simulation results. The simulation exhibits the expected behavior of the ball settling down as the tendons are stretched. (Time advances left to right, top to bottom)

Again, the length of time for the simulation could be dramatically reduced by changing the dynamic model of the strands to avoid the high spring constants and the associated high frequency vibrations that occur in the strands. This is a focus of future work, but our preliminary work in this area suggests that in addition to using strand tension as an input, treating the strands as kinematic variables (while leaving the rest of the hand dynamic) results in between one and two orders of magnitude faster calculation.

6 Conclusions and Future Work

We have developed a strand-based model of the hand that simulates the complete hand. The model is based on variational integrators which provide excellent behavior with constraints, coupling, and closed kinematic chains and fixes the numerical dissipation issues that the hand model in [20] exhibits. The variational integrators also simulate the system directly in generalized coordinates.

This work represents only the first steps towards accurate hand models for manipulation tasks. The next step in modeling is to design more advanced tendon models that are capable of branching, sliding, and becoming slack.

The simulation environment also needs to be extended to handle elastic impacts, plastic impacts, and nonholonomic constraints. Relevant theoretical work has already been done [3] and variational integrators are known to handle these well. `trep` is currently being improved in this direction.

Better models of friction, including stick/slip phenomenon are also needed. This is an active research area in the dynamics community. The progress there is expected to work well in the variational integrator setting.

A hand model can only be as accurate as the parameters used to design it. System identification experiments are needed to collect empirical data and characterize the hand [16]. This includes measuring properties like elasticity or damping, and improving abstract representations like spring networks for the extensor tendons.

Lastly, this paper does not address control for manipulation. Future work in this direction includes automating the calculation of Jacobians for torque/force calculation and linearization of the dynamics for feedback control design (including optimal control).

References

1. B. Baker. The magic touch: Scientist invent the cyberhand, a brain controlled robotic hand with fingers that can actually feel. *http://www.popsci.com/article/2006-03/magic-touch*, 2006.
2. E. Biryukova and V. Yourovskaya. A model of hand dynamics. In F. Schuind, K. An, W. Cooney, and M. G. Elias, editors, *Advances in the Biomechanics of Hand and Wrist*, pages 107–122. Plenum Press, 1994.
3. R. C. Fetecau, J. E. Marsden, M. Ortiz, and M. West. Nonsmooth Lagrangian mechanics and variational collision integrators. *SIAM Journal on Applied Dynamical Systems*, 2003.
4. J. R. Flanagan, P. Haggard, and A. M. Wing. The task at hand. In A. M. Wing, P. Haggard, and J. R. Flanagan, editors, *Hand and Brain*, pages 5–13. Academic Press, Inc, 1996.
5. H. G. F.R.S. *The Complete Gray's Anatomy.* Merchant Book Company, Finland, 2003.
6. M.-C. Hepp-Reymond, J. Huesler, and M. A. Maier. Precision grip in humans: temporal and spatial synergies. In A. M. Wing, P. Haggard, and J. R. Flanagan, editors, *Hand and Brain*, pages 33–68. Academic Press, Inc, 1996.
7. E. R. Johnson and T. D. Murphey. Scalable variational integrators for constrained mechanical systems in generalized coordinates. *IEEE Transactions on Robotics*, Submitted.

8. L. Jones and S. Lederman. *Human Hand Function.* Oxford University Press, New York, NY, 2006.
9. A. Kochan. Shadow delivers first hand. *Industrial robot: an international journal*, 32(1):15–16, 2005.
10. O. Kounchev and M. J. Wilson. *Mathematics of Surfaces.* Springer Berlin / Heidelberg, Bulgaria, 2003.
11. K.-H. Lee and K. H. E. Kroemer. A finger model with constant tendon moment arms. In *Proceedings of the Human Factors and Ergonomics Society*, pages 710–714(5), 2007.
12. J. Lin, Y. Wu, and T. S. Huang. Modeling the constraints of human hand motions. In *Human Motion, 2000. Proceedings. IEEE*, pages 121–126, 2000.
13. R. L. Linscheid. Historical perspective of finger joint motion: the hand-me-downs of our predecessors. *Journal of Hand Surgery*, 27 A:1–25, 2002.
14. J. E. Marsen and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, pages 357–514, 2001.
15. M. G. Pandy. Computer modeling and simulation of human movement. *Annual Review of Biomedical Engineering*, 3:245–73, 2001.
16. J. Pearlman, S. Roach, and F. Valero-Cuevas. The fundamental thumb-tip force vectors produced by the muscles of the thumb. *Journal of Orthopaedic Research*, 22(2):306–312, 2006.
17. J. L. Sancho-Bru, A. Perèz-Gonzalez, M. Vergara-Monedero, and D. Giurintano. A 3-D dynamic model of human finger for studying free movements. *Journal of Biomechanics*, 34:1491–1500, 2001.
18. F. Scheepers, R. E. Parent, W. E. Carlson, and S. F. May. Anatomy-based model of the human musculature. In *International Conference on Computer Graphics and Interactive Techniques, Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 163–172, 1997.
19. M. H. Schieber. Individuated finger movements: rejecting the Labeled-Line Hypothesis. In A. M. Wing, P. Haggard, and J. R. Flanagan, editors, *Hand and Brain*, pages 81–98. Academic Press, Inc, 1996.
20. S. Sueda, A. Kaufman, and D. K. Pai. Musculotendon simulation for hand animation. In *ACM SIGGRAPH conference proceedings*, volume 27, 2008.
21. M. West. Variational integrators. *California Institute of Technology Thesis*, 2004.
22. A. M. Wing. Anticipatory control of grip force in rapid arm movement. In A. M. Wing, P. Haggard, and J. R. Flanagan, editors, *Hand and Brain*, pages 33–68. Academic Press, Inc, 1996.
23. S. Zilber and C. Oberlin. Anatomical variations of the extensor tendons to the fingers over the dorsum of the hand: a study of 50 hands and a review of the literature. *Plastic Reconstructive Surgery*, 113:214–221, 2004.