Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005), pp. 1–10
K. Anjyo, P. Faloutsos (Editors)

Helping Hand: An Anatomically Accurate Inverse Dynamics Solution For Unconstrained Hand Motion

Winnie Tsang, Karan Singh, Eugene Fiume

Department of Computer Science, University of Toronto
{wtsang,karan,elf}@dgp.toronto.edu

Abstract

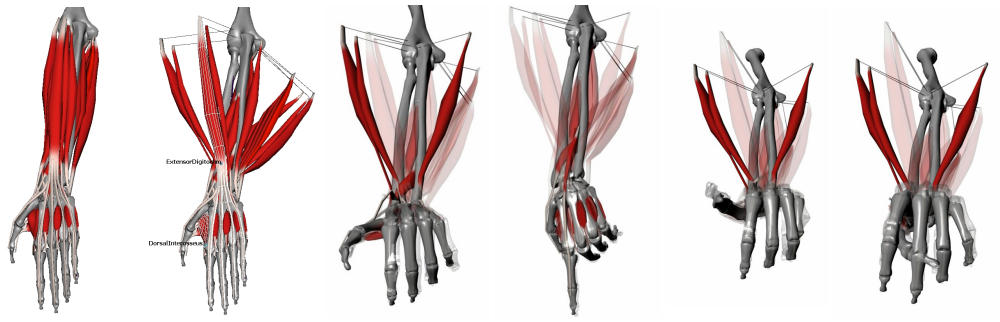
We present a realistic skeletal musculo-tendon model of the human hand and forearm. The model permits direct forward dynamics simulation, which accurately predicts hand and finger position given a set of muscle activations. We also present a solution to the inverse problem of determining an optimal set of muscle activations to achieve a given pose or motion; muscle fatigue, injury or atrophy can also be specified, yielding different control solutions that favour healthy muscle. As there can be many (or no) solutions to this inverse problem, we demonstrate how the space of possible solutions can be filtered to an optimal representative. Of particular note is the ability of our model to take a wide array of joint interdependence into account for both forward and inverse problems. Given kinematic postures, the model can be used to validate, predict or fill in missing motion and improve coarsely specified motion with anatomic fidelity. Lastly, we address the visualization and understanding of the dynamically changing and spatially compact musculature using various interaction techniques.

1. Introduction

The human hand is an exquisitely flexible biomechanical device that has evolved from a multiplicity of function over millennia, beginning with our move to bipedalism, the use of hand-held implements, and the development of verbal, written, gestural and musical communication [MM00]. As with many other marvels of evolution, it is easy to ignore the remarkable utility of our hands–until, that is, we sustain even the smallest injury that inhibits their function. The complexity of the human hand makes the construction of accurate visual and physical models an extremely challenging goal. Such a model is, however, necessary in exacting disciplines such as surgery, the study of hand pathologies, biomechanics, education and paleo-archaeology, and they can greatly benefit expressive character animation.

The exploration of dynamic hand function requires an accurate model that is visually, biomechanically and anatomically valid. As work in computer graphics takes on a deeper collaborative role with fields such as anatomy, surgery, education, archeology and biology, the outcomes of our physical models and simulations will need to be more than visually persuasive. For example, the pathology of a repetitive stress injury likely has and will require a visual manifestation; however, predicting an injury, modeling therapies, or simulating preventative measures, all require a hand model that attends to anatomical and physical validity. That said, a hand model need not be an exact physical replica. This is unlikely to be necessary even if it were possible. Instead we require a synthetic biomechanical model of the hand that can be used to test hypotheses and to operate in a manner that is consistent with the observable physical expectations of hand function. Our goal is to choose the simplest model necessary to meet such task requirements. There are further bounds to physical modeling. A strictly physical model is for example unlikely to predict pain, or the preferred hand position for a guitar chord, or the accepted fingering of a piano piece. We instead contribute a robust model that accurately estimates contraction impulses and physical parameters required by a real human hand to assume a given posture or to perform a certain task, and to allow domain experts on hand function from various disciplines to work with these models and draw conclusions based on their expertise and needs.

The ability to simulate and animate biomechanical hand motion, and to explore muscle activations needed for a hand to perform tasks is the basis of our contribution. We cannot



(a) (b) (c) (d) (e) (f)

Figure 1: *Musculosketal architecture of the hand in (a), is explored by spreading, highlighting and annotating selected muscules in (b).* Clinical application*: (c) and (d) show inverse simulation solutions to motion drills.* Animation application*: (e) and (f) show an inverse solution to the motion capture of the letters "A","S", performed in A.S.L. A ghosted hand shows the fitting error.*

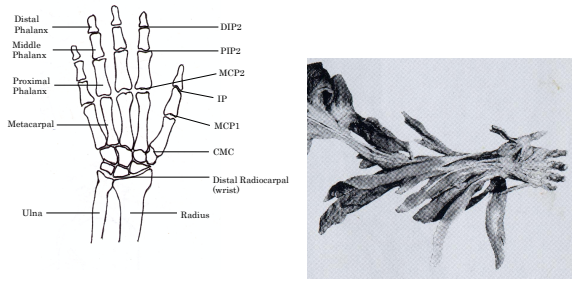
model what we do not know, and there is much that is still not understood about the neuro-muscular control of the human hand. Here perhaps our models may help to close the loop with experimental sciences to improve our overall understanding of hand function through both simulation and visualization.

Our goal is thus to build a skeletal musculo-tendon model of the hand that combines computer graphics, physical mechanics and anatomical and clinical measurements for unconstrained hand motion. We restrict ourselves to unconstrained hand motion so as to avoid attempting to measure and environmental forces and isolate them from forces generated by the hand. Such a model would provide forward simulation capabilities for visualization and pose prediction, and from which we could derive inverse dynamics control solutions for hand animation. This is of particular interest in a clinical setting to explore biomechanical hand function. A realistic, robust inverse dynamic model for a given musculature also allows us to address an increasingly important topic in character animation, one of validation. Just as computer characters today, real or fictional, are modeled with a virtual skeleton for kinematic control, they may in the near future be constructed with an anatomic musculature of the kind we describe in this paper. The musculature could then be used dynamically for simulation, as an interpolation mechanism to fill in coarsely keyframed data, or as a reality check to validate whether the character is capable of performing a given animated motion. An anatomic hand musculature can greatly augment vision and hand motion capture systems to fill in missing motion data resulting from occlusion that is characteristic for complex hand motion.

Our hand model consists of a complex 3D musculoskeletal model with parameterized musculo-tendons accurately attached to skeletal components. To animate a model, our system incorporates anatomical and biomechanical data to perform (forward) simulation; given a time sequence of muscle activations. Hand motion is computed by solving the equations of motion and integrating them in time using a stable implicit integrator. To effect control over a model, we need to solve the inverse dynamics problem, which recovers muscle activations given a set of desired joint orientations. A way is needed to traverse the large space of physical control solutions to find biomechanically valid if not optimal solutions. We discuss several possible techniques below but because we wish to permit a wide range of unrestricted hand motion, we select an optimization approach based on minimizing the error between the target and output joint angles subject to inequality constraints on the parameters of the objective function; those parameters are the muscle activations we seek. Using animations either from simulation or motion capture data, the target joint angles and muscle effort efficiencies are formulated into objective functions, which are then minimized to recover contraction values.

The contribution of our work is a biomechanical, animated human hand model (see Figure 1) containing:

- a comprehensive biomechanically realistic human hand and forearm architecture with real-time physical simulation using rigid body dynamics. Control forces from the actuation of 41 musculo-tendon units around 16 joints produces bone rotation based on mechanical laws and experimental data from studies of human hand anatomy. Joint interdependence is anatomically modeled.
- a constraint based inverse dynamics solution that recovers muscle actuation values for the purposes of analyzing muscle function and hand animation.
- a graphical interface for visualizing the complex anatomical layout of the hand, and the output of the system. Controls are provided for the clinical practitioner selectively visualize and spread muscle while maintaining context and to vary the actuation values of the musculo-tendon units and their strength associated with their capacity to generate force.



(a) Hand joint nomeclature

(b) Plastinated hand ©Gunther von Hagen.

Figure 2: *Hand nomenclature and visualization.*

2. Related Work

Anatomy and Biomechanics. Studies in the anatomy of hand joints, tendons and muscles are the basis for building biomechanical hand models. Computer hand models [AHS03, BY94, Moc96, ES03] are based on the anatomical structure of the hand consisting of a hierarchical arrangement of bones, as can be found in illustrated anatomy books [AL99]. Studies of musculo-tendon configurations provide degree of freedom and their limit for joints, and give insight into joint interdependency. However this interdependency is still poorly understood, and the prevailing view is that both biomechanical and neurological issues are at play.

Extensive studies have been conducted by Brand and Hollister [BH99] to describe hand function. They explore muscle operation, joint co-ordination, and force transmission via tendons. A thorough description of hand muscles with information about average resting fiber length, tension fraction, and moment arms can be found in [BH99]. [NTH01, NTHF02] described different anatomically-based models for physical and geometric reconstruction of muscles. This physical muscle representation incorporates Hill's classic three-element model [Zaj89], which mathematically relates parameterized musculo-tendon forces to actual physiological measurements. [TSBNLF05] address the construction of skeletal muscle geometry from the visible human dataset and its subsequent simulation modeling collision and contact using a robust finite element method.

Delp et al. [DL00] developed computational tools to create models of musculoskeletal system that can be used in combination with experimental studies to answer questions related to a variety of research and clinical applications. For example, their Software for Interactive Musculoskeletal Modeling (SIMM) allows users construct computer models of a variety of musculoskeletal structures which consist of a set of rigid segments connected by joints. Given muscle activations, a mathematical muscle model computes the force and moments that each muscle generates, and the resulting joint motions. Our work focuses on the high-quality anatomical visualization of the human hand using both forward and inverse dynamics.

Recently, motion capture data has been used to compute activation levels of facial muscles [SNF05] using the muscle simulation techniques of [TSBNLF05]. Motion capture has also been datamined by a kinematic hand model [ES03] to generate realistic hand postures that capture the sympathetic finger motion to arbitrarily animated hands.

The most comprehensive computer graphical hand model to date is found in the work of [AHS03] and more recently [KM04]. [AHS03] present a human hand model with an anatomical structure suitable for real-time physical simulation of muscles, together with elastic skin properties. Their approach is focused on modeling the overall visual appearance of the hand by using pseudo-musculo-tendon units which control bone rotation, and geometric muscles which deform the skin. Our work uses a similar hand and muscle force model but is more comprehensive in modeling musculotendon interdependencies: for example, the Extensor Digitorium (see Figure 3) has a strong effect on the interdependencies observed among different fingers. While hand neurophysiology is ill-understood, we also attempt to capture interdependencies by aggregating related musculo-tendon units. As our current focus is on the architecture, dynamics and visualization of musculoskeletal models, we have not yet addressed skin tissue (e.g., [KJP02]), [KM04] presents an approach to the construction of a geometric model of real hands based on the CT scans for use in animation and simulation. We hope to employ such an approach to parametrically resize and calibrate our hand model to variations in muscular capabilities of different hands.

Dynamics. Dynamics simulation is extensively used in robotics, biomechanics and computer animation [AHS03, Bar96, BY94]. In this approach, the motion of objects, namely their position as a function of time, is the outcome of a solution of a set of ordinary differential equations for a discrete time step. Associated with dynamic simulation is the issue of control. In biomechanics, the use of inverse dynamics to compute force from noninvasive measurements of body motions is an area of much interest. Most attempts to quantify muscle forces in humans are based on optimization [Cro78]. Static optimization is the most commonly used method to estimate muscle forces during locomotion [TAD03]. [TAD03] investigated the problem of computing of muscle excitation patterns that produce coordinated movements of muscle-actuated dynamic models. The authors introduced a new algorithm that uses static optimization along with feedforward and feedback controls to drive the kinematic trajectory of a musculoskeletal model toward a set of desired kinematics. Their computed muscle excitations were similar in timing to measured electromyographic patterns, thus improving the feasibility of using detailed musculoskeletal models

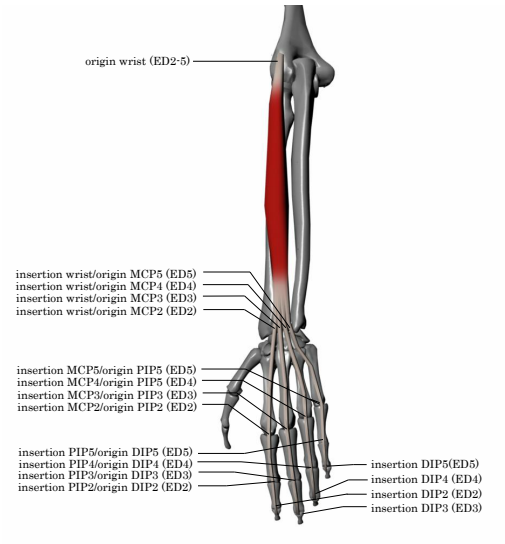


Figure 3: *Extensor Digitorum forking into multiple tendons spanning multiple joints.*

to simulate and analyze movement. Static optimization solves a different optimization problem at each time instant, and is computationally inexpensive. Its main disadvantage is that the results depends on the accuracy of the data recorded. Dynamic optimization is time dependent, and solves a large, often expensive, optimization problem for one complete cycle of the movement. [AP01a] presented a dynamic optimization solution for normal walking on level ground and showed that the predicted muscle forces and joint contact forces of the dynamic and static solutions were remarkably similar.

Visualization. Three-dimensional visualization can present anatomical information in a form that is visually pleasing and more easily understandable. In [HBR*92], volumetric interactive atlases were generated from cross-sectional images. Human anatomy information can be presented as volumetric data. Transparency, segmentation, and deformation are some of the techniques used to display the inside of the volumetric volume. In [MTB03], deformations were used for browsing through data: the user can cut into, open up, spread apart, or peel away parts of the volume in real time, exposing the interior while still retaining surrounding context. This approach has the advantage of not removing potentially important surrounding contextual information.

3. Hand Model

We have seen that there has been considerable research into the depiction of hand motion. However, there to date are no computer-animated, biomechanically valid skeletal musculo-tendon models that can support arbitrary joint interdependencies, forward and inverse dynamics, and aggregate musculo-tendon strain computation. The components of our hand model are: a joint hierarchy, a skeleton consisting of 29 bone meshes, and a set of 41 musculo-tendon units attached to this skeleton. We now elaborate on these components.

Joints. There are 16 joints in our hand model. Joints are modeled as hinges that are capable of rotation about their principal axes. Referring to Figure 2a, there are 23 degrees of freedom in the joint system. Each finger has four, with two DOF for the MCP joint for flexion/extension and adduction/abduction, and one DOF each for the PIP and DIP of joints for flexion/extension. The thumb has five: one for the MCP joint, one for the IP joint, and three for the CMC joint. Finally there are two DOF for the rotation of the wrist. We introduced a extra degree of freedom for the CMC joint of the thumb because the two axes of rotations are not orthogonal. The flexion/extension of the CMC joint occurs in the trapezium, and its abduction/adduction occurs in the first metacarpal bone. See [BH99] for a deeper discussion.

Bones. There are 29 bone meshes in total in our hand model. The bones in the forearm and hand are arranged hierarchically, as are most limbs in the body. As with [BY94], we group bones into 16 different links that join the 16 joints in the articulated hand model (Figure 2a). The sizes and masses of each link were gathered from biomechanical sources [BH99, BY94] and measured from a 3D hand skeleton model. The data for the bone are from averaged values obtained in medical literature [AL99, BY94].

Muscles. There are in total 41 hand and forearm musculotendon units in our hand model. To understand the distinction between muscle and musculo-tendon unit, consider the Flexor Digitorum Superficialis (FDS), Flexor Digitorum Profundus (FDP), and Extensor Digitorum (ED). These muscles originate near the proximal radiocarpal joint (elbow) and the tendons pass through the MCP and PIP joints of the digits 2-5 (index finger through to pinky). Each muscle thus contains *four* musculo-tendon units corresponding to each tendon. The interdependency of related musculotendon units of each such muscle due to neuro-muscular control of the hand produces sympathetic finger motion. This interdependency is ill-understood; it may vary among individuals and thus cannot be easily quantified. After consultation with anatomists, we discovered that in general, the musculotendon units of FDS can be contracted more independently than FDP and ED, which are deep muscles. Thus, we coupled the four musculo-tendons units of FPD and ED so that they are contracted simultaneously, while the four musculotendons of FPS units can be contracted independently.

Biomechanics provides models to quantify musculo-tendon force generation for locomotion. A basic model is the linear spring-damper, which views muscles as spring-like, exhibiting linear forces in the direction determined by an origin and an insertion point. However, a single spring-damper cannot capture multiple parameters taken from actual muscle and tendon anatomical measurements. Hill's three-element model provides a simple, parameterizable muscle repre-

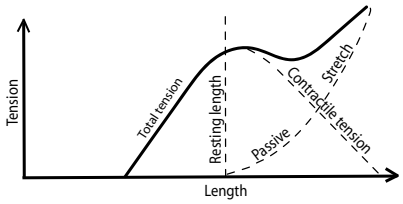


Figure 4: *Blix curve*

sentation onto which actual measurements can be mapped [Zaj89].

There are two ways for a muscle to exert tension: by active contraction due to nerve stimulus (active contraction), and by the muscle's elastic behavior (passive contraction), which is independent of nerve stimulus. We refer to the time-varying neural control signal as the *contraction* or *actuation value* of the muscle, denoted by $c(t)$. In active contraction of a muscle, the highest tension is produced when the muscle is at its resting length. This relationship is summarized in length-tension curves for passive contraction and active contraction, known as the *Blix* curve as shown in Figure 4. We adopt the quadratic curve fit used by [AHS03] to approximate the two length-tension curves, giving as a continuous function the magnitude of the force exerted by a muscle.

We define the current fiber length as $l \in [0.5l_o, 1.5l_o]$, with rest length l_o. The contraction value is $c \in [0,1]$. Let F_{max} be the maximal isometric force that can be exerted by a muscle; it can be thought of as relative maximum strength of the muscle [BBT81]. The contraction force term corresponding to the contractile element in Hill's model is

$$F_c(l) = \left[1 - 4\left(\frac{l}{l_o} - 1.1\right)^2\right] F_{max} \; . \tag{1}$$

The stretch force term corresponds to Hill's parallel element:

$$F_s(l) = \begin{cases} 2.77(l/l_o - 1)^2 F_{max} & l \geq l_o. \\ 0 & l < l_o. \end{cases} \tag{2}$$

The total force is thus

$$F(l) = cF_c(l) + F_s(l). \tag{3}$$

Hill's three-element model only computes the magnitude of the muscle's force, and not its direction. Force direction is computed along its line of action, which can be represented as a sequence of piecewise linear segments as it extends through one or more joints via tendon. We estimated the origin and insertion points for every muscle at every joint it crosses. This introduces some possible inaccuracies. One is that real insertion and origin attachment regions are areas, not points. This is acceptable when the real attachment region is small, as in the thread-like tendons through the wrist and fingers. The muscle origins on the forearm, however, are larger. In principle, we should distribute the lines of action over the insertion area. However, there is no experimental data that provides the distribution of muscle effort over an area of muscle attachment. A piecewise linear approximation also neglects inter-muscle collision forces as adjacent muscles exert force on one other. The pennation angle, which relates the orientation of muscle fibers to tendon tissue can also be used to approximate the force applied to the tendon by the muscle.

To estimate the location of origin and insertion attachment points, we manually fit digitized muscle fibers and tendon data to our 3D skeleton model using *Maya*. We estimated the origins and insertions to be the centers of the muscle or tendon attachments to the bones (see Figure 3).

To address how forces displace bone and joints, let the point at which the tendon attaches to the bone from the joint's center of rotation be the *lever* arm. The lever arms of the muscles as they cross the affected joints are derived from [BH99]. The total effect of a muscle at a joint is given by its torque, which is the cross product of the lever arm and the linear force. The magnitude of the torque is the perpendicular distance between the axis of the joint and the tendon as it crosses the joint, called the *moment* arm. The moment arm determines the effectiveness of muscle tension at a joint. As a tendon crosses multiple joints, it may change direction after crossing one joint on its way to another. However, the tension is the same along the length of the tendon, and is not shared or divided between joints. At each joint, the moment arm will be different, determining the relative effect of a muscle at each joint.

4. Dynamic Simulation

We now discuss the equations of motion of our hand model. Joints have degrees of freedom only in orientation and not in position, since they cannot be translated unless they are skeletally dislocated. Let $\vec{S}_i$ be the state vector for a single rigid body i at time t. Let $\vec{\theta}_i$ be its spatial rotation and let $\vec{\omega}_i$ be its angular velocity. The state of joint i is represented by

$$\vec{S}_i(t) = \begin{pmatrix} \vec{\theta}_i(t) \\ \vec{\omega}_i(t) \end{pmatrix} \; . \tag{4}$$

To compute the total torque at joint i, we sum the torques from every muscle that crosses joint i. Let $\vec{r}_j$ denote the lever arm and $\vec{F}_{i,j}$ denote the force of the muscle j on joint i. We introduce a set of weights, $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_M), \alpha_i \in [0,1]$, to weight the contribution to the total tension that a muscle can exert at time t. These weights model the effect of muscle fatigue. For instance, a muscle with a weight of zero cannot exert tension, whereas a muscle with a weight of one exerts tension normally. If m muscles cross joint i, then its total torque, $\vec{T}_i$ at time t is given by

$$\vec{T}_i(t) = \sum_{j=1}^{m} \vec{r}_{i,j} \times \alpha_j(t) \vec{F}_{i,j}(t) \; . \tag{5}$$

We define $\hat{F}_{i,j}$ to be a unit vector giving force direction, and let $\vec{i}_{i,j}$ and $\vec{o}_{i,j}$ be respectively the insertion and origin points for muscle j at joint i. The force $\vec{F}_{i,j}$ can be computed by

$$\vec{F}_{i,j}(t) = [c_j F_{c,j}(l_j) + F_{s,j}(l_j)]\hat{F}_{i,j} , \quad (6)$$

where

$$\hat{F}_{i,j} = \frac{\vec{i}_{i,j} - \vec{o}_{i,j}}{||\vec{i}_{i,j} - \vec{o}_{i,j}||} . \quad (7)$$

Let $\mathbf{I}_i$ be a 3×3 inertia tensor for joint i, and k_μ be the coefficient of friction. The coefficient of friction is 0.015, taken from [AHS03], which they used in accordance to medical literature. The second-order ordinary differential equation of motion for joint i is given by

$$\vec{T}_i(t) = \mathbf{I}_i(t)\frac{d}{dt}\vec{\omega}_i(t) + k_\mu \vec{\omega}_i(t) . \quad (8)$$

We solve the equations of motion using a stable integration method, namely the implicit Euler method [PTVF02], formulating a system of equations in the form of $\mathbf{A}\vec{x}=\vec{b}$, where

$$\begin{aligned} \vec{x} &= \Delta\vec{\omega}, \\ A &= \frac{\mathbf{I}_t}{\Delta t} + k_\mu, \\ \vec{b} &= \vec{T}_t - k_\mu \vec{\omega}_t. \end{aligned} \quad (9)$$

We can solve the above for $\Delta\vec{\omega}$ by LU decomposition. Once $\Delta\vec{\omega}$ is found we can update the state vector as follows:

$$\vec{\omega}_{t+1} = \vec{\omega}_t + \Delta\vec{\omega}, \quad (10)$$
$$\Delta\vec{\theta} = \vec{\omega}_{t+1}\Delta t, \quad (11)$$
$$\vec{\theta}_{t+1} = \vec{\theta}_t + \Delta\vec{\theta}. \quad (12)$$

We also compute the new length of muscle j as

$$l_j = l_j - \sum_{i=1}^{n} \Delta\vec{\theta} \cdot \vec{r}_{i,j} \quad (13)$$

where

$$l_j = \begin{cases} 1.5 \cdot l_o & , l \geq 1.5 \cdot l_o \\ 0.5 \cdot l_o & , l \leq 0.5 \cdot l_o, \end{cases} \quad (14)$$

clamps muscle length to lie within the range $0.5 \cdot l_o$ to $1.5 \cdot l_o$.

To compute $\mathbf{I}$ effectively, we need to look to the geometry of the hand. For the purposes of computing moments, all joints but the radiocarpal joint (wrist) can be approximated as a sequence of rigid cylinders or rods of constant circular cross-section. This reduces the inertia tensor to a moment of inertia with an analytic solution for each joint approximated in this fashion [Pai02, AHS03]. For the wrist, we assume a constant inertia tensor and compute it once prior to simulation. The inertia tensor is computed using the inertia tensor equation for a particle system, by summing up the squared distances of the bone mesh vertices to the rotation axis multiplied by the mass of bone. Putting all this together, we achieve the forward dynamics simulation as depicted in Figure 5.

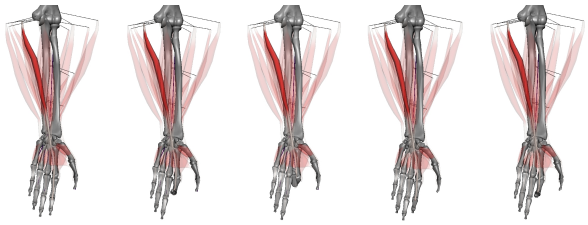


Figure 5: *Forward simulation sequence due to muscle contraction (a)-(c). Same simulation with reduced muscle contraction levels (d),(e).*

5. Constraint-based Control

The technique of using energy constraints on parameterized models was introduced in [WFB87]. Constraints are expressed as *energy functions* on the model's parameter space, which are non-negative functions with zeroes at the points satisfying the constraints. A scalar function over the parameters is obtained by summing these energy functions, and the objective is to find a set of model parameters that minimize the scalar function.

In the forward model, the outputs are the rotation angles and the angular velocities for n joints, and the inputs are m muscle contraction values. Our control algorithm is to recover the contraction values given a sequence of desired rotation angles and angular velocities, which could originate from keyframed animation or motion-capture data. With the recovered contraction values, we can forward simulate the model to produce animated joint movements that closely match target poses. Thus consider the forward simulation as a function f that maps a set of m contraction values to $3n$ rotation angles, $\vec{\theta}_i$, and $3n$ angular velocities $\vec{\omega}_i$:

$$\vec{y} = f(\vec{c}) = \begin{pmatrix} \vec{\theta}_1 \\ \vec{\omega}_1 \\ \vdots \\ \vec{\theta}_n \\ \vec{\omega}_n \end{pmatrix} , \quad (15)$$

where the contraction values for m muscles is $\vec{c} = (c_1, c_2, \ldots, c_m)$, and the function output $\vec{y}$, is a $6n \times 1$ vector, since $\vec{\theta}_i$, $\vec{\omega}_i$ are each 3×1 vectors. Let the targets be $\vec{t} = (t_1, t_2, \ldots, t_{6n})$ for the n joints. We formulate the *energy function* or the *objective function* as

$$E(f(\vec{c})) = w_s \frac{1}{2} E_s(\vec{y}) + w_c \frac{1}{2} E_c(\vec{c}) \quad (16)$$

This is a weighted sum with scalar weights w_s and w_c of the terms E_s and E_c which respectively evaluate the motion, and the controller. By minimizing the motion evaluation term E_s,

we seek to minimize the distance to the desired goals. Minimizing the controller evaluation term E_c promotes a preference for contraction values with lower amplitudes. Mathematically, these energy function terms are expressed as

$$E_s(\vec{y}) = ||\vec{t} - f(\vec{c})||_2^2 , \quad (17)$$

$$E_c(\vec{c}) = \frac{1}{2}||\vec{c}||_2^2 , \quad (18)$$

where $||\cdot||_2$ denotes 2-norm of a vector. Thus we see that E_s constrains the states of the model by measuring the error between the output of the simulator and the target. E_c maximizes the efficiency of the controller by summing the amplitudes of the contraction values.

Since the energy function and its gradient are the weighted sum of its constraints, adding more constraints to the system is simple. The energy function is solved by minimizing the sum of the nonnegative objective functions associated with every goal:

$$E(\vec{c}) = \sum_i w_i E_i . \quad (19)$$

Recall from Eq. 15 that the forward dynamics operation f is a nonlinear function of m muscle contraction values that are all in [0,1]. For optimization, we use L-BFGS-B, a limited memory, iterative algorithm for solving large nonlinear constrained or unconstrained optimization problems [ZBLN97].

More than twenty tendons cross the wrist joint, making it difficult to search interactively for a control solution that simultaneously minimizes all joint errors. We thus first minimize the objective function by omitting the wrist joint. Once a solution for all joints except the wrist is found, we invoke the equations of motion with the candidate joint contraction values to evaluate the resulting error in the wrist joint relative to the target values. We then formulate the objective function to minimize this error; however, we only include as degrees of freedom a restricted set of seven muscles that only affect the wrist joint. Specifically, these muscles are *Flexor Carpi Radialis* (FCR), *Palmaris Longus* (PL), *Flexor Carpi Ulnaris* (FCU), *Extensor Carpi Radialis Longus* (ECRL), *Extensor Carpi Radialis Brevis* (ECRB), *Extensor Carpi Ulnaris* (ECU), and *Abductor Pollicis Longus* (APL), which act as wrist stabilizers. Finally, we combine the resulting seven muscle contraction values with the previously solved ones and forward simulate the system. The sequence of control activations performed by our nervous system on the muscles controlling our wrist and fingers is still unknown. It is thus possible that although our method of decoupling the wrist joint works well in practice, it may not be biologically valid.

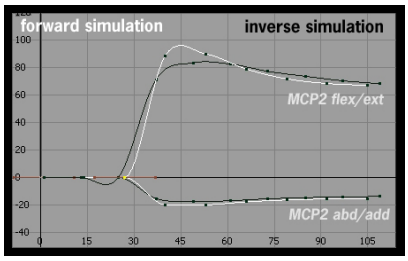
This approach to the musculoskeletal hand control problem is the first we have seen in the literature, and it has some useful properties. First, it recognizes that while the forward simulation problem is deterministic, the inverse problem is more complicated: there can be many different control solutions to achieve the same motion. While this is true of most inverse problems in animation, the solution space for our control system is of extremely high dimensionality. Our two-step approach has to date given us very good results. Second, it is possible to create different affinities for muscle activations; as such our control method is able to search for different solutions in the case, for example, of muscle fatigue, pathology or atrophy. Third, because our physical model captures crosstalk across muscles and joints, the control solutions will naturally exhibit this effect.

To search for different solutions, we used various clinically motivated heuristics for *Repetitive Strain Injuries* (R.S.I.) diagnosis to explore the solution space by simulating muscle fatigue by means of setting muscle weights and solving again for an alternative solution. The starting point for the solutions in Figure 6(c)-(f) is the same.

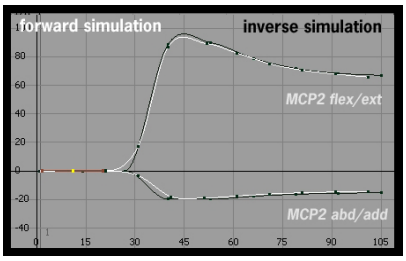
User-controlled. Musclo-tendon unit weights can be manipulated by the user to disable or damp the contribution of units to the solution. Figure 6(a) and (b) shows animation curves of the inverse solutions to the forward simulated motion of flexing the MCP index joint. Comparing the results of performing inverse simulation with all of the muscles enabled, and with only the active muscles plus a small subset enabled, the latter gives a more accurate match to the target motion.
Top-usage. This approach makes use of a previous solution to find the musculotendon units that are most used, by summing up their contraction values throughout the entire animation sequence. Thus the primary muscles will be assigned a smaller weight in the inverse simulation (see Figure 6(d)).
Thresholding. This approach also makes use of previous solutions to simulate fatigued and stretched or pulled muscles by searching for musculo-tendon units with their contraction values or lengths exceeding set thresholds. Thus, musculotendon units that are stretched, pulled or contracted past their thresholds are assigned a smaller weight (see Figure 6(e)).
Random Combination. In this approach we randomly select an user-defined number of musculotendon units and assign a smaller weight to them (see Figure 6(f)).

6. Visualization

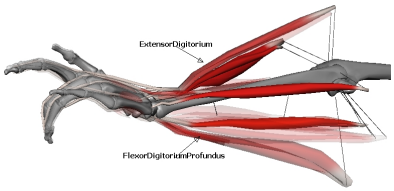
We have implemented a variety of visualization techniques to depict hand animation (see Figures 1 and Figures 7). We illustrate muscle activity through the use of opacity, providing a less obscured view of actively contracting muscles. Muscles also bulge under tension, which is visually easier to perceive than variations in lengths, providing compelling feedback on muscle contraction. We visualize errors between the inverse solution and target motion data through ghosting. The complex and compacted muscle architecture of the hand makes it difficult to visualize the deep structures. Adopting from [AL99], we categorized the forearm musculotendon units into six groups: three groups for the anterior aspect and three groups for the posterior aspect of the forearm; those group are divided based on depth. Each group can be ”spread" by positioning the associated controller objects. This ”spreading" is inspired from Von Hagen’s plasti-



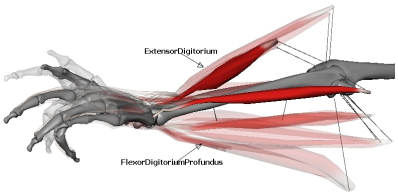
(a) Animation curve of inverse solution for MCP2 using all muscles



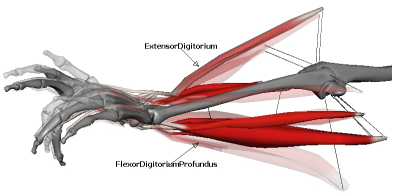
(b) Animation curve of inverse solution for MCP2 using a superset of active muscles



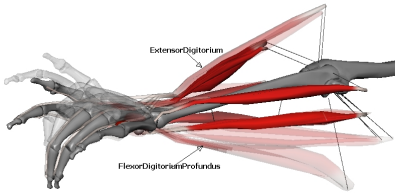
(c) Best Fit Solution



(d) Solution without primary muscles

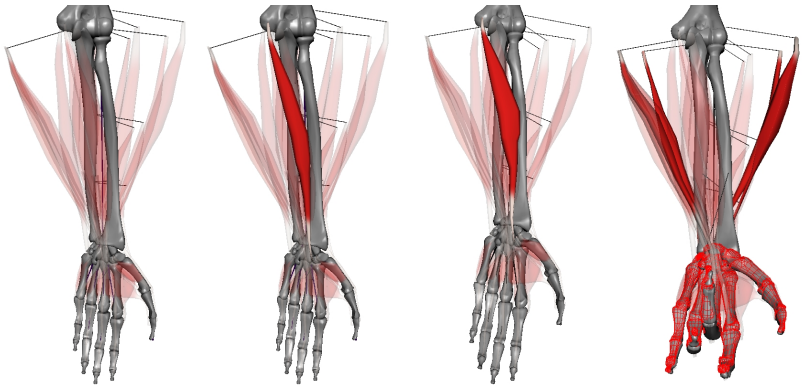


(e) Solution without fatigued muscles



(f) Randomly weighted solution

Figure 6: *Exploring inverse solution space.*



(a) Uncontracted muscles spread within context

(b) Muscle contraction controlling transparency

(c) Contracted muscle controlling bulging

(d) Ghosted skeleton indicating simulation error

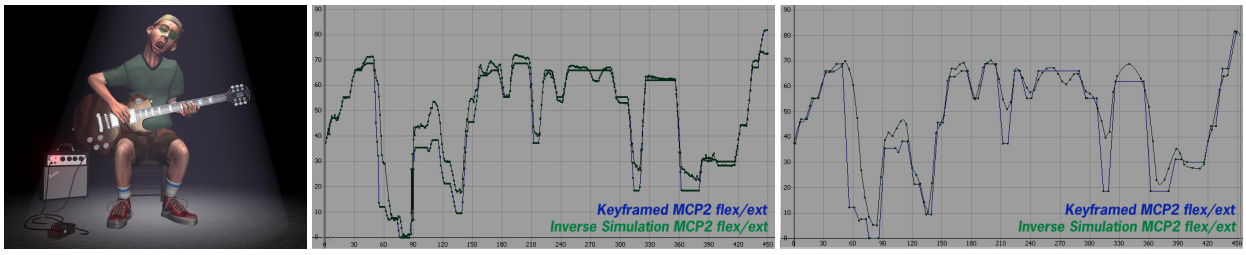
Figure 7: *Various techniques used to aid dynamic muscle visualization in a clinical application.*

nation works (see Figure 2b), which similar to peeling muscle from its attachment. Our goal through these visualization techniques is to present hand architecture and motion to clinicians and animators in ways that illuminate hand function.

7. Implementation and Results

Our hand is composed of three modules: the forward simulator, the inverse simulator, and the muscle force model. all implemented within the commercial animation system *Maya*.

The forward simulator takes the muscle activation values and weights as input, which are defined by the user by keyframing the muscles' attributes using the *Maya* animation interface. It performs implicit integration of the equation of motions described above for each user-defined frame step, starting and ending at the frames specified by user. We have found a time step of to 24 frames per second allows for stable integration. The forward simulator produces finger configurations for each frame step, and the resulting animation can be viewed by advancing the time slider in the interface.



(a) Handrix animation (b) Simulation matching each frame of the final keyframed motion (c) Simulation matching only the keyframed postures

Figure 8: *Simulating a hand keyframed using motion capture based IK postures*

The inverse simulator takes finger configurations as input, which are joint rotations; as an option angular velocities can also be used. It solves for muscle activation values, producing finger configurations that conform to the input for each frame step. The frame step is defined by the user, and should be chosen according to the number of frames that separate the keyframed target data. Sources of target data include keyframed animation, motion captured data, or animation from the forward simulator.

The muscle's force model defines the computation of the force exerted by the muscle on each joint. The input to the module is the activation value and the output is the resulting torques at each affected joint. The modularity of our architecture allows for replacement of the muscle's physical model with a more sophisticated one.

7.1. Clinical Application

Helping Hand provides anatomists with tools for visualizing the anatomical structure of hand muscles and function. Using the forward simulation component, they can manipulate activation values to obtain the resulting motion for contracting particular muscles. Using inverse simulation, anatomists can examine which muscles are utilized to produce unconstrained hand motion. The redundancy of muscles allows different sets of muscles to be used to give similar motion. This redundancy is essential since in the case of muscle fatigue or failure, other muscles can be used as temporary replacement or reinforcement. Hand surgeons are also concerned with the inability of patients to move certain joints due to various pathologies. Anatomists can manipulate muscle weights to find optimal solutions over various groups of muscles. Figure 5d,e shows the result by changing the muscle weights on a set of muscles with similar function.

We recorded a male and female subject performing various American sign language gestures as input to our inverse simulation component. Using such data, we can validate and calibrate our hand model by observing the errors in the inverse solution. Figure 1(e,f) shows the result of the inverse simulation fit to hand motion capture data. The errors for the digits were quite small with moderate wrist error. Wrist stabilization remains a challenging biomechanical problem: while qualitatively quite satisfactory, fully validating the results requires an intrusive deep tissue scan which is unethical for human subjects.

7.2. Animation Application

Helping Hand can also be used to validate and improve the quality of synthesized hand animation. Keyframed hand poses were linearly interpolated producing hand animation which ignores hand dynamics, since biological motion is nonlinear in nature. Using this data as target motion data, we obtained an inverse solution that matches every frame of the keyframed motion. This results in a jittery hand animation due to the difficulty of the hand model to match the non-realistic, linear motion. We then performed another inverse simulation matching only the keyframed postures, producing hand animation that is less robotic and more realistic than the keyframed animation. The motion curves for the flexion/extension of MCP index joint using both approaches are shown in Figure 8 for comparsion.

8. Conclusion and Future Work

The goal of this paper was to present a complete framework for the anatomic modeling, simulation and visualization of dynamic hand motion. Overall, our system takes a first step toward the graphical understanding of hand function. Much work remains to be done. The representation, interaction and visualization aspects of the musculoskeletal data are more mature than the algorithmic modules for hand simulation. The model requires greater refinement and validation against clinical data. The control solution needs to be more flexible, and it may require revision should the clinical case for the two-step control solution (decoupling wrist stabilisation) be proved to be an erroneous assumption. The hand and forearm need a good skin model. Our near-term aim is to work with clinical practitioners on the deployment of our hand model in medical practice. There is also considerable room for optimism that our hand anatomy can be evolved in reverse to develop families of hand models and morphologies for our ancestors, given fossil evidence.

References

[AHS03] ALBRECHT I., HABER J., SEIDEL H.: Construction and animation of anatomically based human hand models. In *Proceedings of SIGGRAPH Symposium for Computer Animation 2003* (2003), vol. 22, ACM Press / ACM SIGGRAPH, pp. 98–109. 3, 5, 6

[AL99] AGUR A. M. R., LEE M.: *Grant's Altas of Anatomy*, 10 ed. Lippincott Williams & Wilkins, Baltimore, Maryland, USA, 1999. 3, 4, 7

[AP01a] ANDERSON F. C., PANDY M. G.: Dynamic optimization of human walking. *Journal of Biomechanical Engineering 123*, 3 (2001), 381–390. 4

[Bar96] BARAFF D.: Linear-time dynamics using lagrange multipliers. In *Proceedings of SIGGRAPH 1996* (1996), vol. 22 of *Computer Graphics Proceedings, Annual Conference Series*, ACM Press / ACM SIGGRAPH, pp. 137–146. 3

[BBT81] BRAND P., BEACH R., THOMPSON D.: Relative tension and potential excursion of muscles in the forearm and hand. *Journal of Hand Surgery 6*, 3 (1981), 209–219. 5

[BH99] BRAND P., HOLLISTER A.: *Clinical Mechanics of the Hand.*, 3 ed. Mosby - Year Book, Inc., St. Louis, MO, 1999. 3, 4, 5

[BY94] BIRYUKOVA E., YOUROVSKAYA V.: A model of hand dynamics. In *Advances in the Biomechanics of Hand and Wrist* (1994), Plenum Press, New York, pp. 107–122. 3, 4

[Cro78] CROWNINSHIELD R.: Use of optimization techniques to predict muscle forces. In *Trans. of the ASME* (May 1978), vol. 100, pp. 88–92. 3

[DL00] DELP S. L., LOAN J. P.: A computational framework for simulation and analysis of human and animal movement. vol. 2, pp. 46–55. 3

[ES03] ELKOURA, G., AND SINGH, K. 2003. Handix: Animating the human hand. In *In Proceedings of the 2003 ACM SIGGRAPH/ Eurographics Symposium on Computer animation*, vol. 22, 110–119. 3

[HBR*92] HOHNE K., BOMANS M., RIEMER M., SCHUBERT R., TIEDE U., LIERSE W.: A volume-based anatomical atlas. *IEEE Comput. Graph. Appl. 12*, 4 (1992), 73–78. 4

[KJP02] KRY P., JAMES D., PAI D.: EigenSkin: Real time large deformation character skinning in hardware. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer animation* (July 2002), pp. 153–159. 3

[KM04] KURIHARA T., MIYATA N.: Modeling deformable human hands from medical images. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), ACM Press, pp. 355–363. 3

[MM00] MARZKE M., MARZKE R.: Evolution of the human hand: approaches to acquiring, analysing and interpreting the anatomical evidence. *Journal of Anatomy 197* (2000), 121–140. 1

[Moc96] MOCCOZET L.: *Hand Modeling and Animation for Virtual Humans*. PhD thesis, Univ. of Geneva, 96. 3

[MTB03] MCGUFFIN M., TANCAU L., BALAKRISHNAN R.: Using deformation for browsing volumetric data. In *Proceedings of IEEE Visualization* (2003), IEEE Computer Society Press, pp. 401–408. 4

[NTH01] NG-THOW-HING V.: *Anatomically-Based Models for Physical and Geometric Reconstruction of Humans and Other Animals*. PhD thesis, Univ. of Toronto, 2001. 3

[NTHF02] NG-THOW-HING V., FIUME E.: Application-Specific Muscle Representations. In *Proc. Graphics Interface* (May 2002), pp. 107–116. 3

[Pai02] PAI D.: Strands: Interactive simulation of thin solids using cosserat models. *Proceedings of Eurographics 2002, Computer Graphics Forum 21*, 3 (September 2002), 347–352. 6

[PTVF02] PRESS W., TEUKOLSKY S., VETTERLING W., FLANNERY B.: *Numerical Recipes in C++, 2nd Edition*. The Press Syndicate of the University of Cambridge, Cambridge, UK, 2002. 6

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *Proceedings of SIGGRAPH 2005*(2005), ACM Press. 3

[TSBNLF05] TERAN J., SIFAKIS E., BLEMKER S., NG-THOW-HING V., LAU C., FEDKIW R.: Creating and Simulating Skeletal Muscle from the Visible Human Data Set. *IEEE Transactions on Visualization and Computer Graphics*, (May/June 2005), vol. 11, 317–28. 3

[TAD03] THELNE D. G., ANDERSON F. C., DELP S. L.: Generating dynamic simulations of movement using computed muscle control. vol. 36, pp. 321–28. 3

[WFB87] WITKIN A., FLEISCHER K., BARR A.: Energy constraints on parameterized models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), ACM Press, pp. 225–232. 6

[Zaj89] ZAJAC F. E.: Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering 17* (1989), 359–411. 3, 5

[ZBLN97] ZHU C., BYRD R. H., LU P., NOCEDAL J.: Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw. 23*, 4 (1997), 550–560. 7