

HELPING HAND: AN ANATOMICALLY ACCURATE INVERSE DYNAMICS
SOLUTION FOR UNCONSTRAINED HAND MOTION

by

Winnie Tsang

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Copyright © 2005 by Winnie Tsang

Abstract

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

Winnie Tsang

Master of Science

Graduate Department of Computer Science

University of Toronto

2005

The human hand is a marvelously useful biomechanical device that has inspired numerous depictions and models, however its complexity, makes the construction of accurate visual and physical models an extremely challenging goal. We present, Helping Hand, a realistic skeletal musculo-tendon model of the human hand and forearm. The model permits direct forward dynamics simulation, which accurately predicts hand joint orientation 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 can also be specified, yielding different control solutions that favor healthy muscle. The model can take kinematic pose data and predict animation sequences of the hand to fit or indeed physically improve upon kinematic data. Lastly, we demonstrate ways to isolate and visualize muscle groups as their effect on hand motion is animated.

Acknowledgements

I would like to thank my supervisor, Karan Singh, for his guidance, assistance, and support. I love his approach to supervision, letting me work at my own comfortable pace, and providing a nice and friendly work environment. I would also like to thank Eugene Fiume, for all his help and great feedback. Thank you so much for finding time to read this thesis.

I would like to thank Dr. Nancy McKee, and Dr. Anne Agur, from the Faculty of Medicine, for educating me about the anatomy and biomechanics of the hand. For the longest time, I couldn't understand the abduction/adduction of the thumb, and Anne was so patient with me.

Also, thanks to everyone in the DGP lab for being so approachable and nice. I would like to thank Aaron Hertzmann for his help with the numerical optimization aspect of this work; Michael J. McGuffin for his help with the visualization aspect; and John Hancock for his patience and his time in assisting me with motion capture system. In addition, thanks to Abhishek Ranjan, Bowen Hui, Minnie Hui, Alexander Kolliopoulos, Eron Steger, Jack Wang, Mike Wu, for the animes, movies, card-games, dinners, and the great time I had at DGP.

Finally, thanks to my family, Dad, Mom, and Tony, for their love and support. Especially to Tony for his help with Maya, and his help with the modeling of those muscles.

This work was financially supported through a Post Graduate Scholarship from the Natural Science and Engineering Research Council.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Approach	4
1.3	Contribution	5
1.4	Thesis Organization	6
2	Previous Work	7
2.1	Anatomy and Biomechanics	7
2.2	Hand Models In Computer Science	9
2.3	Character Animation Techniques	11
2.3.1	Forward Kinematics and Inverse Kinematics	11
2.3.2	Dynamic Simulation	12
2.3.3	Visualization	16

3	Anatomy and Biomechanics of the Hand	18
3.1	Anatomy	18
3.1.1	Hand Bones	19
3.1.2	Hand Muscles	19
3.2	Biomechanics	20
3.2.1	Hill Three-Element Muscle Model	20
3.2.2	Muscle line of action	26
4	Anatomical Hand Model	30
4.1	Models of Joints	30
4.2	Models of Bones	32
4.3	Models of Muscles	32
5	Dynamic Simulation	35
5.1	An Unconstrained Rigid-Body System	35
5.2	Integration of Ordinary Differential Equations	37
5.2.1	Explicit Euler Method	38
5.2.2	Implicit Euler Method	39
5.2.3	Runga-Kutta Method	39
5.3	Dynamic Simulation of Hand Model	40

6	Control: Constraint-based approach	43
6.1	Energy Constraints of Hand Model	43
6.2	Minimization of Function	45
6.3	The Constrained Simulation Loop	46
6.4	Exploring the Solution Space	48
7	Visualization	51
7.1	Visualization Techniques	51
7.1.1	Transparency	52
7.1.2	Muscle Deformation	52
7.1.3	Spreading	53
7.1.4	Ghosting	55
8	Implementation and Results	56
8.1	Modularity of the System	56
8.2	Application and Results	59
8.2.1	Clinical Application	59
8.2.2	Animation Application	62
9	Conclusion	66

9.1	Future Work	67
9.1.1	A More Sophisticated Hand Model	67
9.1.2	Validation and Calibration of Hand Model	69
9.1.3	Visualization	70
A	Hand Data	71
A.1	Hand Joints and Bones Data	71
A.2	Hand Muscles Data	75
B	Inertia	82
B.1	The Inertia Tensor	82
B.2	The Transformation of Inertia Tensor	83
B.2.1	Rotation	83
B.2.2	Translation	84
B.3	The Inertia Tensor of Joints	85
C	Gradient Calculation	88
C.1	Gradients of Energy Functions	88
	Bibliography	91

List of Tables

3.1	Functions of Extrinsic Muscles	28
3.2	Functions of Intrinsic Muscles	29
A.1	Average Range of Motion for Hand Joints	73
A.2	Lengths, Radius, Thickness, and Mass of Hand Bones	74
A.3	Hand Muscles Fiber Lengths and Tension Fractions	75
A.4	Hand Muscles Lever Arms	76
A.5	Muscles Origins and Insertions co-ordinates	79

List of Figures

3.1	Anatomical grouping of forearm muscles	21
3.2	Hill's three-element model	22
3.3	Blix curve	24
3.4	Extensor Digitorum forking into multiple tendons spanning multiple joints. . .	27
4.1	Hand joint nomenclature	31
6.1	Exploring inverse solution space	50
7.1	Muscle contraction controlling transparency	52
7.2	Muscles bulge under tension	53
7.3	Exploring hand musculoskeletal architecture	54
7.4	Ghosting for indicating simulation error	55
8.1	System architecture	57

8.2	Screenshot of the Helping Hand scene implemented in <i>Maya</i> 's animation platform	58
8.3	Interface for keyframing musculo-tendon units' contractions.	60
8.4	Forward simulation sequence with non-weakened musculo-tendon units	61
8.5	Forward simulation sequence with weakened muscle	61
8.6	Clinical application of inverse simulation	63
8.7	Simulating a hand keyframed using motion capture based IK postures	65

Chapter 1

Introduction

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. While the full set of factors affecting its development and function still await explication, it appears that the evolution of the human hand, with its opposable thumb and its lightweight, dextrous fingers, was interleaved with the move by our ancestors to bipedalism, their growing use of hand-held implements, their need to grip variably-sized objects, and the development of verbal, written, gestural and musical communication. An exquisitely flexible biomechanical device emerged from such a multiplicity of function [MM00].

The exploration of dynamic hand function, morphology, evolution and pathology requires an accurate model that is visually, biomechanically and anatomically valid, and not simply physically based. Because we often make judicious assumptions and simplifications in computer graphics to keep the focus on visual modeling, we would be quite fortunate if the forces and torques computed by most physically based models were even to have actual physical or biomechanical validity. However, as computer graphics works with other disciplines, 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 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, in addition to our traditional requirement that the model be suitable for computer animation.

Furthermore, there are bounds to physical modeling. It would be very difficult, for example, to develop a synthetic hand model that predicts which hand injury is more painful than another. A strictly physical model is also unlikely to predict the preferred hand position for a guitar chord, or the accepted fingering of a piano piece. However, we can develop a model that accurately computes the energy required (or strain induced) for 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 compute the muscle activations needed for a hand to perform tasks is the basis of our contribution.

We cannot model what we do not know, and there is much that is still not understood about 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.

In this thesis, we address the problem of developing and controlling a biomechanical hand model that incorporates empirical data from anatomical studies of hand joints, hand tendons and hand muscles. In contrast to most works in computer graphic hand models that focus

on controlling joint angles to generate animations of hand movement, instead we use these animations to discover how the hand functions in the muscular aspect. We also present a solution to the inverse problem of determining an optimal set of muscle activations to achieve a given pose or motion. In addition, there is a problem of redundancy associated with the hand model, since multiple muscles can perform similar functions at a joint, for example, to flex joint. This redundancy poses an interesting question: if certain muscles are weakened or removed, is it still possible to produce the same movement? As a matter of fact, muscle fatigue is a common issue, caused by the physiological and biochemical events. It leads to a decline in the capacity of a muscle to generate force, and during its recovery, other muscles must be relied upon. By specifying muscle fatigue, injury or atrophy, we can yield different control solutions that favor 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.

1.1 Motivation

Kinematic and dynamic techniques have traditionally provided character animators with high-level skeletal control. Used alone, kinematics is insufficient in dealing with hand motion, since inverse kinematics (IK) algorithms typically deal with constraints along a single chain, whereas a multi-appendage, limb-like human hand requires complex constraints among joints of different chains. In dynamics, a character is modeled using a spring-damper system of rigid bodies and muscles. The control problem associated with dynamics can be resolved using PD controllers. However this disregards the complex interaction among muscles and joints.

In medical science, computer graphical models are effective tools for visualizing human movement, displaying musculoskeletal configuration, and analyzing forces in motion for educational and surgical purposes. Combining the principles of mechanics, and measurements from hand anatomy, biomechanical models are developed to analyze muscle function, study movement

abnormalities, design new medical products, and guide surgery.

Our goal is 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. Such a model would provide forward simulation capabilities for visualization and pose prediction. This is useful in understanding the functionality of each hand muscle. We could derive inverse dynamics control solutions for hand animation. This is of particular interest in a clinical setting to explore biomechanical hand function. To verify our biomechanical model, we present the system with both synthetic data from simulation and motion capture data, and minimize the error between the resulting motion from the solutions with the presented ones. In addition, through a realistic dynamic model, keyframed animation data can be evaluated and improved upon by using it as the initial conditions to the inverse dynamics problem.

1.2 Approach

Helping Hand can be divided into three parts: modeling, animation, and control. In this section, we present a brief description of each part. In subsequent sections of the document, a detailed description of the techniques, and the method of implementation are given.

Our hand model consists of a complex 3D musculoskeletal model with parameterized musculo-tendons accurately attached to skeletal components. It is composed of 16 hand joints, with a total of 24 degrees of freedom, and 41 musclo-tendon units. The bones are modeled as rigid bodies connected by joints, and musclo-tendon units are attached to the bones at its origin and insertion points.

To animate the model, our system incorporates anatomical and biomechanical data to perform (forward) simulation; given a time sequence of musclo-tendon units activations. We use rigid

body dynamics subject to control forces of musculo-tendon units. External forces from the environment are ignored. We wish to keep the system simple enough for real-time simulation, and the equations of motion simple enough such that its derivative can be found symbolically. Hand motion is computed by solving the equations of motion and integrating them in time using a stable implicit integrator. By specifying musculo-tendon unit activations at each discrete time step, one can observe the functionalities of each musculo-tendon unit in isolation or in combination.

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. This problem is similar to the common issue of control associated with physical systems. Several different techniques can be used to address this control issue, for example, using a constraint-based approach [IC87, PB88, WFB87, WK88], or building hand-crafted controllers [Mil88, TT94]. Since we can formulate our problem in terms of optimization, we have selected a constraint-based approach. We minimize 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.

1.3 Contribution

To summarize, the contribution of our work is a biomechanical, animated human hand model 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 produce 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 to vary the actuation values of the musculo-tendon units and their strength associated with their capacity to generate force.

1.4 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 presents previous work in anatomy and biomechanics, in computer graphics, and in dynamics relevant to the human hand. Chapter 3 and Chapter 4 explores the anatomy and biomechanics of the hand and presents our biomechanical hand model. Chapter 5 discusses the concepts of rigid bodies dynamics, and mathematical formulation of the equations of motion for our hand model. Chapter 6 provides an overview of the constraint-based approach and its application in our hand model. Chapter 7 describes the visualization techniques, and Chapter 8 shows the implementation details, applications and results. Chapter 9 provides conclusions and possible extensions to our work.

Appendix A collects all relevant data from studies in the anatomy and biomechanics of the human hand and presents them in tabulated format. Appendix B explains the calculation of the inertia tensor, and Appendix C illustrates the mathematical formulation of the gradient of the objective function to the minimization problem.

Chapter 2

Previous Work

Research in biomechanical hand model and hand motion spans several areas of interest, including anatomy and biomechanics, character animation techniques including forward/inverse kinematics, dynamic simulation, and motion capture. The problem of controlling the dynamic system, so that the system will move toward the desired state, must be dealt with. In this chapter, we present an overview of the work in these areas as they relate to our work.

2.1 Anatomy and Biomechanics

Studies in the anatomy of hand joints, hand tendons and hand muscles provide the basis for building biomechanical models of the hand. Computer hand models [AHS03, BY94, GE03, Moc96, MTA⁺01] 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 in the musculo-tendon configuration provide joint limits and the degrees of freedom associated with each joint, and they give insights into joint interdependency. However this interdependency is still poorly understood, and the prevailing view is that both biomechanical and neurological

issues are at play. Until more information is revealed regarding the the muscular and neurological aspects of the hand, a complete hand model cannot be developed.

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 their average resting fiber length, tension fraction, and moment arms can be found in Brand et al. [BH99]. Although their book is directed to surgeons planning a tendon transfer operation, it contains valuable empirical information that can be used to parameterize a biomechanical muscle model to capture the features of a particular muscle.

Ng-Thow-Hing and Fiume [NTH01, NTHF02] described different anatomically-based models for physical and geometric reconstruction of muscles. Physical representation of the muscles are mathematical models that compute musculo-tendon forces with parameters that relate to actual physiological measurements of musculo-tendons. One such model that allows empirical measurements to be used in order to capture the features of any skeletal muscle is the Hill's three-element model [Hil38]. In this model, the musculo-skeleton is composed of a contractile element, a parallel element and a series element. The series element represents the elastic effects of the tendon and the restorative forces that the tendons generate when they are stretched. The parallel element and contractile element represents the passive and the active elastic properties of muscle respectively. In our model, we used Brand and Hollister's measurements to parameterize Hill's three-element model to represent each individual muscle-tendon unit.

Hoy et al. [HZG90] developed a musculoskeletal model for computer simulation studies of musculo-tendon function and muscle coordination during movement. Although their work is for the human lower extremity, similar muscle models can be employed for hand muscles.

Biryukova et al. [BY94] constructed a biomechanical model which include 29 principal muscles and degrees of freedom of the hand. Muscles are simplified to be threads with their origins

and insertions as points. The model approximates real hand dynamics, and an inverse dynamics approach is used to solve for muscle efforts produced during coordinated hand movements. Our model differs in that we isolated muscles that extend via tendons to multiple digits as individual musculo-tendon units. Modeling a muscle as separate musculo-tendon units allows us to capture the effect of different areas of a muscle being simulated, which plays an important role in producing the sympathetic motion between digits. Furthermore, instead of concerning with muscle efforts required for a given hand movement, we solve for activation values, incorporating neurology of the hand into our model.

Delp et al. [DL00, TAD03] 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, Software for Interactive Musculoskeletal Modeling (SIMM) [DL00], lets users construct computer models of a variety of musculoskeletal structures which consist of a set of rigid segments connected by joints. Given muscle activation, a mathematical model of muscle computes the force and moments that each muscle generates, and the resulting joint motions.

2.2 Hand Models In Computer Science

Hand models in computer graphics, like most articulated figures, are modeled as a collection of rigid bodies connected by joints with one or more degrees of rotational freedom. Most of the effort in computer animation of the human hand is concerned with the ability to grasp objects, and with the modeling of skin deformation associated with the motion [Moc96, MTLT88]. In [MTLT88] *Joint-dependent Local Deformations* (JLD) are proposed for hand deformations. This model is based on two structural layers: skeleton and geometric skin. JLD operators define a initial coordinate system relative to the initial skeleton for each point on the geometric skin, and also define a final coordinate systems relative to the final skeleton. Each point is then

deformed by a change of reference system, producing a new skin that fits the final skeleton. In [Moc96] Dirchlet free-form deformations (DFFDs) are tailored for hand deformations. Early work on grasping [RG91] introduced the commonly used joint angle constraint $\theta_{DIP} = \frac{2}{3}\theta_{PIP}$ to approximate the interdependence of hand joints. [LK95] proposed a model that included dependence constraints between DIP and PIP joints of each finger and among MCP joints of the rest of the fingers; this does not capture the interdependencies that exist among the DIP and PIP joints of different fingers.

Vision researchers solve the reverse problem of computer graphics, and have employed simplified hand models for image-based gesture recognition. [MTA⁺01] presents a anatomical human hand model capable of producing gestures in American Sign Language. They used a forward kinematic approach with a simple set of interactive controls for the positioning of fingers.

Recently, motion capture techniques have generated considerable interest [KGP02, PB02]. This work shows how motion capture data can be segmented and used to synthesize new motion, without the numerical complexity involved in simulation and controlling physical-based system, and less tedious than keyframing. Kovar et al. [KGP02] presents a method for managing motion capture data by automatically constructing a directed graph, termed as *motion graph*. The motion graph consists of clips of the original motion and automatically generated transitions between those motion clips. Thus realistic motion can be synthesized by finding a path on the graph that meets the user's specifications. [PB02] describes a technique of enhancing keyframed animation with motion captured data. Their technique is comprised of two processes: texturing and synthesizing. By texturing, details are added to degrees of freedom that were keyframed, and by synthesizing, motion are created for degrees of freedom that are not keyframed. Motion Capture Data has also been used for hand motion synthesis. Elkoura and Singh in [GE03], adopt a similar data-driven approach to generate realistic hand motion by using motion captured data to add sympathetic finger motion to arbitrarily animated hands.

The most comprehensive computer graphical hand model to date is found in the work of Albrecht et al. [AHS03]. They developed a human hand model with an anatomical structure suitable for real-time physical simulation of muscles, together with elastic skin properties. They use a hybrid muscle model consisting of 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. However, their model does not capture the interdependencies of musculo-tendon units that belong to the same muscle: for example, the Flexor Digitorum Profundus which has a strong effect on the interdependencies observed among different fingers. While hand neurophysiology is ill-understood, we attempt to capture the set of possible interdependencies by aggregating related musculo-tendon units. We have also not yet added a skin tissue layer to our model and have instead concentrated on building a system capable of mapping contraction values to bone positions, and the reverse procedure of mapping bone positions to contraction values using optimization techniques.

2.3 Character Animation Techniques

Character animation techniques includes kinematics and dynamics approaches. We will discuss briefly how they are used in human model positioning.

2.3.1 Forward Kinematics and Inverse Kinematics

Forward kinematics is a method of controlling an articulated object by explicitly defining its state vector at a specified time. The state vector is given by $\mathbf{S} = [S_1, \dots, S_n]$, where S_1, \dots, S_n describes the orientations and positions of all the joints in the articulated model. The joints are linked in a hierarchical manner, such that changes in the state of one joint along the chain get propagated down to the end, known as the end-effector. Let \mathbf{X} be the the state of the end-

effector, $\mathbf{X} = f(\mathbf{S})$, and is defined by a functional mapping from the all the transformation of joints up the hierarchy. This method is used in keyframed animation where the animator specifies the states at keyframes and the computer interpolates the object's state between these keyframes. The interpolation forms a trajectory curve, and research focused on manipulating the shape of the trajectory for increased smoothness of generated motion. For highly articulated figures, animating with this technique becomes increasingly difficult. Despite its limitation, this technique provides an animator full control of the states at the keyframes, and thus its use is prevalent in the animation community.

Inverse kinematics [ZBLN97] is associated with procedural animation where the kinematics is determined based on implicit instructions rather than explicit positions. It is the reverse of forward kinematics, where the user defines the desired state of the end-effector and the algorithm computes a solution that positions and orients all the joints up the chain. Inverse Kinematics has difficulties in handling systems with many degrees of freedom. They also have trouble when the problem becomes under-constrained. This technique allows precise positioning of the end effector, and has been used in human model positioning [ZBLN97].

2.3.2 Dynamic Simulation

Dynamics simulation is extensively used in robotics, biomechanics and computer animation [AHS03, BW92, Bar96, BY94, GT95, TT94, Wit01]. In this approach, an object's motion is governed by laws of physics. Objects have physical properties such as mass, inertia, and energy, and their states are found by solving a set of ordinary differential equations at discrete time step. There are various techniques available for integration of ordinary differential equations, such as Euler integration, implicit Euler integration, Runge-Kutta Method, etc, [PTVF02] to recover the states.

Simulating the physical behavior of objects can produce more realistic motion. However this

approach may involve very complex sets of differential equations that are too computationally intensive for real-time applications. A considerable research effort has been devoted to the acceleration of physical simulation. Grzeszczuk et al. [GTH98] proposed a new approach to creating physically realistic animation that exploits neural networks to emulate physical dynamics that are trained off-line through the observation of the physics based model in action. Fang et al. [FP03] described a method for efficient synthesis of highly dynamic motions by using a set of objective functions and constraints that lead to linear time analytical first derivatives resulting in fast per-iteration computation times.

The Control Problem

Associated with dynamic simulation is the issue of controlling the dynamic system. There is a family of force-based constraints method that compute forces to cancel parts of the applied forces that act against the constraints. They include:

- *Dynamic Constraints*: use inverse dynamics to find forces that fulfill the constraints.
- *Reaction Constraints*: supply reactive forces that offset forces violating constraints and the add forces that fulfill the constraints.
- *Penalty Method*: add extra energy terms to the minimization problem that penalize violations of constraints.
- *Augmented Lagrange Multipliers*: add differential equations to the system that causes the systems to fulfill multiple constraints. This approach is more suitable for a flexible model since often more than one constraints must be satisfied.

Constraints can also be expressed with respect to energy, space, or time. [WFB87] formulated constraints as energy functions, and a search is performed in the model's parameter space in the direction specified by the energy gradient, which is the derivative of the energy function

with respect to the parameters. Through optimization, a solution is found where the energy function approaches zero indicating that the constraints are satisfied. Isaacs and Cohen [IC87] in their system, DYNAMO (for dynamic simulation of linked figures), used inverse dynamics to determine the forces required to perform a specified motion. Spacetime constraints were introduced by Witkin and Kass [WK88], where constraints on the motion were specified as: what the character must do, how the action is performed, the physical structure and properties of the character, and the resources available to the character. These constraints combined with Newton's law can be formulated as a constrained optimization problem, where the solution is a physically valid motion that satisfies the "what" criteria and optimizes the "how" criteria.

Inverse dynamics, to compute force from noninvasive measurements of body motions (position, velocity, and acceleration of each segment), is a heavily researched in the biomechanics community. Since the number of muscles crossing a joint is greater than the number of degrees of freedom specifying joint movement, the force developed by each muscle cannot be determined uniquely. Most attempts to quantify muscle forces in humans are based on the application of optimization theory. Static optimization is the most commonly used method to estimate muscle forces during locomotion [AP01b, TAD03]. This method is time independent, solving a different optimization problem at each time instant, and is computationally inexpensive, even when applied to very detailed models of the body. The main disadvantage of static optimization is that the results depends on the accuracy of the data recorded during a motion analysis experiment, specifically the positions, velocities, and accelerations of the body segments. The difficulty in estimating velocity and acceleration from position measurements, leads to significant errors in the calculated values of the net joint torques and, therefore, in the estimates of muscle force. Furthermore, it is difficult to include muscle physiology in the formulation of a static optimization problem because estimates of muscle length and contraction velocity also depend on the accuracy of the measured data. Whereas, dynamic optimization [AP01b, AP01a] is time dependent, and solves one optimization problem for one complete cycle of the movement, and thus more computationally expensive. The system equations are

integrated forward in time, and so the problem may be formulated independent of experimental data. Muscle physiology can also be easily incorporated in the formulation of the problem.

[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 to simulate and analyze movement. In [AP01a], they presented a dynamic optimization solution for normal walking on level ground. They used experimental gait data to specify the initial and final states of the simulation, formulating a constrained optimization problem. [AP01b] compared the two methods of optimization, static and dynamic. They showed that the predicted muscle forces and joint contact forces of the dynamic and static solutions were remarkably similar.

In general, constraint based approaches formulate the control problem as an optimization problem with the goal of minimizing or maximizing an objective function over a time interval. The objective function is nonlinear and requires the use of expensive numerical optimization techniques. Most efficient numerical optimization techniques require the computation of the objective function's gradient to formulate the search direction. Therefore, this optimization approach is restricted in application to fairly simple physical models such as those where their objective functions' gradients can be symbolically computed.

Another method of controlling a dynamic system is by motion synthesis. This requires the construction of controllers to generate control functions for a set of actuators to drive the dynamic model. For instance, the pattern of muscle actuation of snakes and worms was modeled as sinusoidal waveforms in [Mil88]. Hand-crafted controllers have been developed for both rigid, articulated figures, as well as deformable models [TT94]. The process of controller synthesis is difficult and often tedious. To automate the process, optimization techniques that minimize

a control objective function through repeated forward dynamic simulation and motion evaluation have been used [GT95, GTH98]. In [GTH98], Grzeszczuk presents the NeuroAnimator which approximates a physical system with neural networks trained through examples, and subsequently uses the trained network for control synthesis by adjusting the actuation values through back-propagation of the network error. The system has the advantage of producing physically valid motion with less computation cost than integration of equations of motion. In addition, the neural network is differentiable, thus the effect of control forces on the actions of the model can be easily computed for controller synthesis. This is useful for complex systems, where the differentiation of the equations of motion is too difficult or computationally expensive. The drawback of this system is that large amounts of examples must be provided in training the network, and the off-line training process can be time consuming.

2.3.3 Visualization

Three-dimensional visualization can present human anatomy information in a form that is visually pleasing and easily understandable. Computerized anatomic atlases have the advantage of providing various interactive approaches to anatomical information and education. Direct interaction allows for a better understanding of anatomical structure in spatial relation to their surroundings. This is an ideal complement to conventional cadaveric dissections. In addition, 3D reconstruction of anatomical structures provides a virtual environment which can realize surgical planning, virtual surgery, virtual endoscopy, and training simulations. In [HBR⁺92] volume based 3D interactive atlases have been generated from cross-sectional images. These atlases are based on a two layer model of image volumes linked to a semantic network containing descriptive knowledge. Human anatomy information can be presented as volumetric data, and many traditional techniques for visualizing the inside of this data involve removing portions of the data to reveal the interior. However, this approach has the disadvantage of removing potentially important surrounding contextual information. Transparency, segmenta-

tion, and deformation are some of the techniques used to display the inside of the 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.

Chapter 3

Anatomy and Biomechanics of the Hand

To appreciate the dynamics of the real hand, it is important to gain a basic understanding of the key concepts in anatomy and biomechanics. In this chapter, we explore these two fields of study - anatomy and biomechanics, focusing on work that is relevant to hand modeling.

3.1 Anatomy

Anatomy is a field of science that deals with the structure of the body [AL99, NTH01]. Bone, ligaments, tendon, muscles and nerves and connective tissue together define the body's form and motion. Bones make up the skeleton that defines the structure of the articulated body. Ligaments bind the bone-ends together, and guide motion to prevent dislocation and excessive movement that might cause breakage. Muscles provide forces that are transmitted via tendon to the bone to generate motion. Finally, nerves are transmitters of nerve impulses that cause skeletal muscle fibers to contract, thus producing force. In the following subsections, we will focus in particular on the anatomical layout of hand bones and muscles and the functions of individual hand muscles.

3.1.1 Hand Bones

The bones in the forearm and hand are arranged in a hierarchy as with most limbs in the body [AL99]. In the forearm, there are two bones, *radius*, and *ulna* that are located between the proximal radioulnar (elbow) joint and distal radioulnar joint (wrist). The palm of the hand consist of eight carpal bones: *hamate*, *capitate*, *trapezoid*, *trapezium*, *scaphoid*, *lunate*, *pisiform*, and *triquetrum*. In addition, the palm has five metacarpal bone for each digit of the hand. Of particular interest is the *trapezium* that forms a saddle-shaped joint with the 1st metacarpal bone, allowing the thumb to be opposed, which is essential to human survival. From the metacarpal bones, three phalanx bones are attached in this order: proximal, middle, and distal that form the fingers. For the thumb, there are two phalanx bones, proximal and distal. Refer to [AL99] for diagrams that illustrate this hierarchical arrangement.

3.1.2 Hand Muscles

Muscles of the hand can be grouped according to their locations and their functions [AL99]. They can be categorized into intrinsic (originate in hand) and extrinsic (originate in forearm). The intrinsic muscles are responsible for delicate finger movements, and the extrinsic muscles are responsible for flexing and extending of fingers and stabilizing the wrist.

For the extrinsic muscles, the interosseous membrane that connects the radius and ulna serves as a separator of muscle groups. The extrinsic muscles on the volar side of the membrane are said to be in the anterior compartment, and generally flex or pronate the hand. They can be subdivided into three groups: superficial, middle, and deep. The extrinsic muscles on the dorsal side of the interosseous membrane are said to be in the posterior compartment, and generally extend or supinate the hand. Figure 3.1 illustrates this grouping. The intrinsic muscles can be grouped according to the digit they move. Thenar muscles affect the thumb, hypothenar

muscles the 5th or little finger, and lumbricals and interossei affect digits 2-5. The main actions of each muscle is tabulated in Table 3.1, and Table 3.2.

3.2 Biomechanics

From the field of biomechanics, models are available to quantify how muscles and tendons exert force for locomotion. One such model is the *linear spring-damper* muscle model that simplifies muscle to be spring-like. Springs which exhibit linear forces in the direction determined by an origin and an insertion point. The force of the muscle can be calculated by:

$$f^m = k_s(l^m - l_o^m) - k_d \frac{dl^m}{dt}, \quad (3.1)$$

where k_s is the stiffness coefficient and k_d is the damping coefficient of the spring. l^m denotes the length of the muscle, and l_o^m is its rest length. This representation has worked well with periodic motions that are inherently sinusoidal, but they may not be suitable for a larger class of motions that require sudden changes of velocity and may require more nonlinear elasticity and damping in the dynamics. Biomechanists need a model that has parameters corresponding to empirical muscle and tendon measurements to capture the features of any skeletal muscle in a body. *Hill's three-element model* provides such a parameterizable representation.

3.2.1 Hill Three-Element Muscle Model

To understand how muscles exert tension, we consider active contraction due to nerve stimulus and the muscle's elastic behavior, which is independent of nerve stimulus. To model the way muscles exert tension, Hill's model has three elements [Hil38, NTH01, Zaj89]: the series element (SE), the parallel element (PE), and the contractile element. The series element and the parallel element represent the tension exerted by the muscle's elasticity, and the contractile

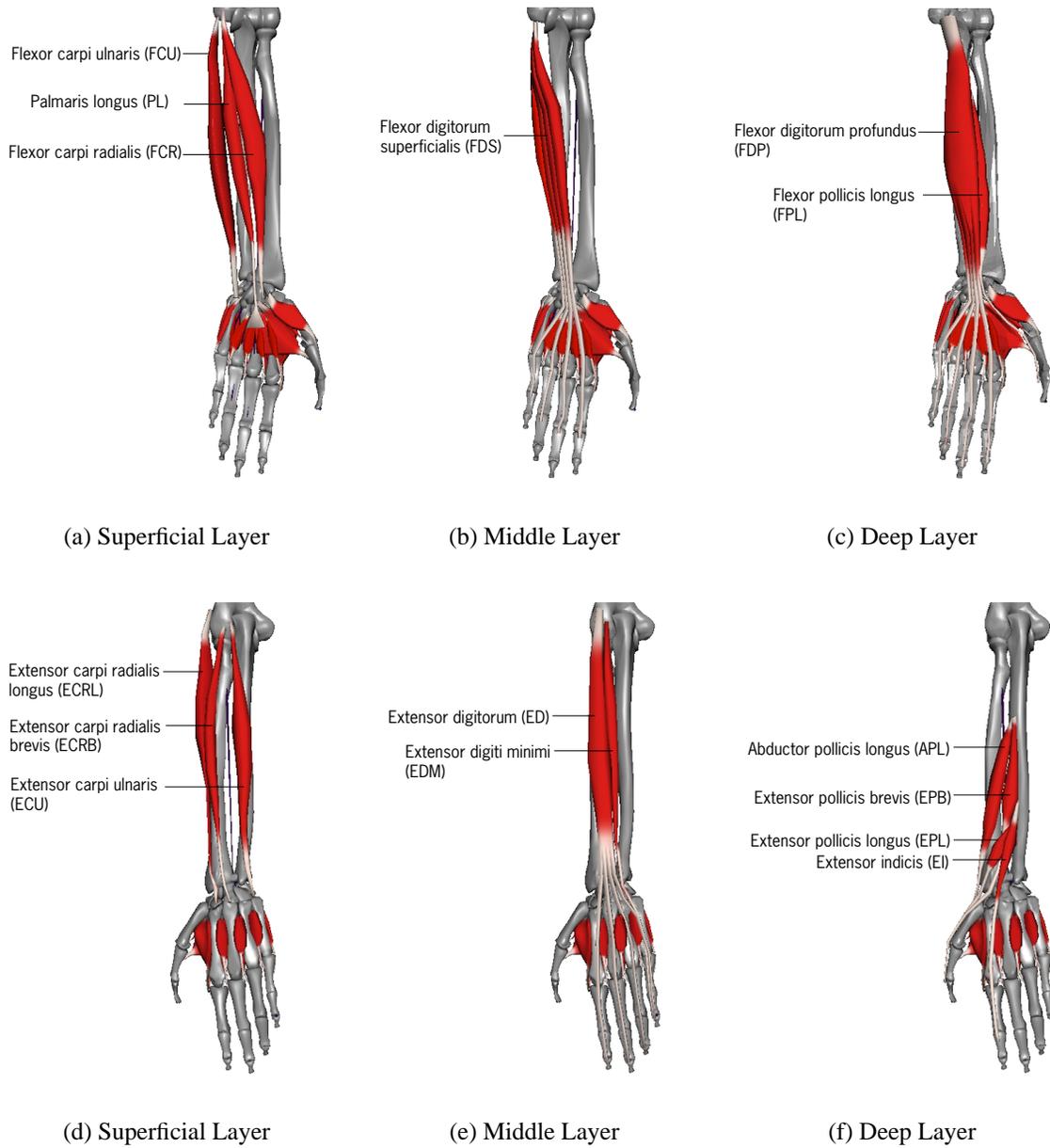


Figure 3.1: Anatomical grouping of forearm muscles: (a) - (c) Muscles on the Anterior Surface of the Forearm, (d) - (f) Muscles on the Posterior Surface of the Forearm

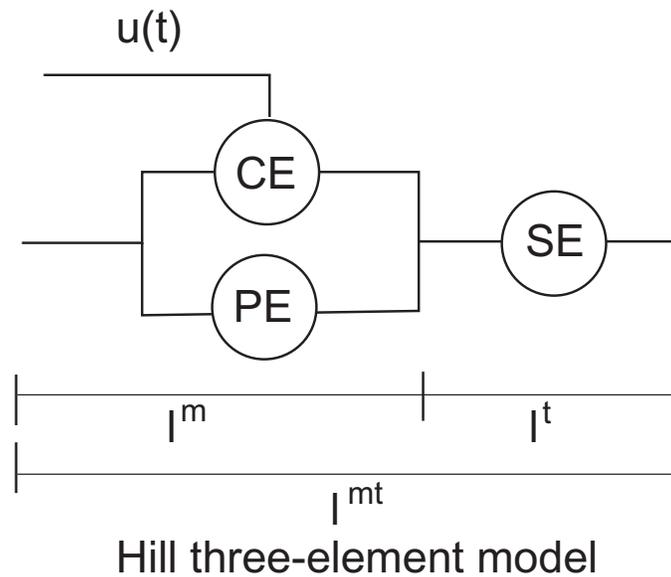


Figure 3.2: the Hill three-element model consisting of the contractile element (CE), parallel element (PE), and series element (SE).

element represents the tension exerted by active contraction. In the following subsections, each of the elements will be further discussed in detail.

Series Element

The series element, in series with the contractile element, captures the series elasticity of the muscle. Series elasticity refers to the elastic effects of the tendon and the small adaptability that allow sarcomeres to shorten slightly during isometric contraction of the muscle. The tension produced by the series element can be approximated with a simple quadratic function that is more suitable for simulation purposes than other more complex forms:

$$f^{SE}(l^{SE}) = \begin{cases} k \cdot (l^{SE} - l_o^{SE})^2, & l^{SE} \geq l_o^{SE} \\ 0, & l^{SE} < l_o^{SE}. \end{cases} \quad (3.2)$$

The descriptor SE refers to the series element. l^{SE} , and l_o^{SE} is the length of the tendon and the slack length of the tendon respectively. Notice that tendon only produce tension when they are stretched. Since tendon only lengthen to a small extent as the muscle is shortened, such that its length is fairly constant, we neglect the tension from the series element in our biomechanical hand muscle model.

Parallel Element

The parallel element, in parallel with the contractile element, captures the parallel elasticity of the muscle. *Parallel elasticity* refers to the elastic tissue that stretches when the muscle elongates and recoils when the muscle shortens. Tension is produced when the muscle is stretched beyond its rest length, referred to as *passive contraction*, which is independent of nerve stimulus. The length-tension curve for passive recoil is shown in Figure 3.3.

We approximate the length-tension curve for passive contraction by a simple quadratic function, where l^m denotes the length of the muscle, and l_o^m is its rest length:

$$f^{PE}(l^m) = \begin{cases} k \cdot (l^m/l_o^m - 1)^2, & l^m \geq l_o^m \\ 0, & l^m < l_o^m. \end{cases} \quad (3.3)$$

Contractile Element

The contractile element is associated with force generation from active contraction controlled by neural control signal. In this paper, we refer to this time-varying neural control signal as the *contraction/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 or isometric length. By combining the two curves for passive contraction and active contraction, we get the *Blix* curve as shown in Figure 3.3.

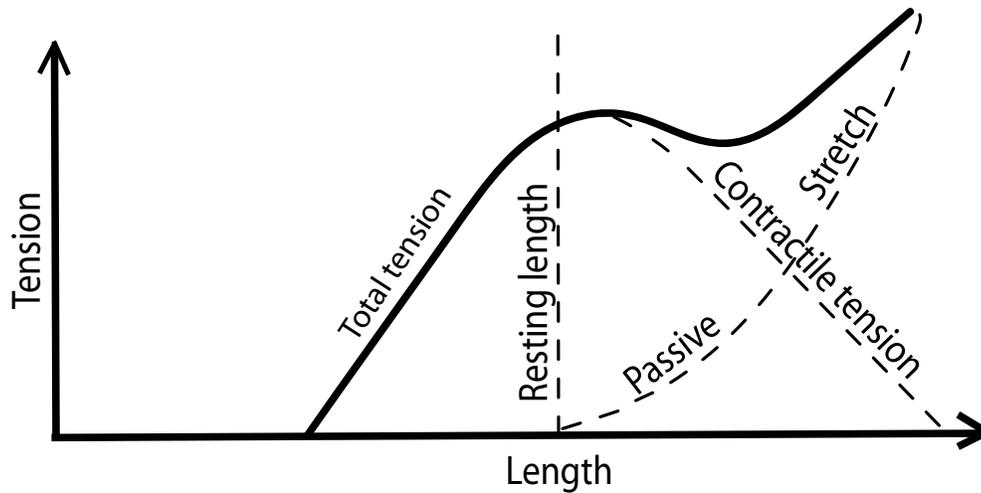


Figure 3.3: *Blix curve* The normalized version of the muscle's force-length curve for passive and active contraction

Through experiments, the force exerted by the contractile element can be summarized by two important properties: *force-length* and *force-velocity*. Force-length property relates the muscle force to its length. The force-length function is given by:

$$f^{CE}(l^m) = f_o^m \cdot \left[1 - \frac{l^m - l_o^m}{W} \right]^2. \quad (3.4)$$

The parameter W controls the width of the concave parabolic curve. The lower ranges of the curve at which force is zero, are where the normalized length ($\frac{l^m}{l_o^m}$) is equal to 0.5 and 1.5. Again, l^m denotes the length of the muscle, and l_o^m is its rest length or isometric length. The maximum isometric force is denoted by f_o^m . Isometric contraction happens when there is tension on the muscle but no movement is made, causing the length of the muscle to remain the same. f_o^m is different for each muscle, since muscles are capable of exerting varying amount of tension depending on the density and length of the fibers. f_o^m can be estimated as the relative potential excursion of muscles, and is documented in [BBT81]. Potential excursion is the distance through which a muscle can contract actively, and if a muscle length is shortened past this distance, it can no longer exert any tension. The potential excursion is proportional to the mean fiber length, and the work capacity of a muscle is proportional to its volume of muscle

fibers. Thus, by dividing the volume by fiber length, the physiological cross-sectional area (PCSA) of the muscle is determined, which is proportional to the relative tension.

$$f_o^m = 25 \frac{N}{cm^2} \cdot PCSA. \quad (3.5)$$

The force-velocity property is the relationship between muscle force and its velocity of shortening. The velocity of muscle contraction is inversely proportional to the force. A large force cannot be exerted in very rapid movements, and that greatest velocities are attained under conditions of low loading. Hill's hyperbolic equation is as follows, where v^m is the velocity of muscle shortening:

$$f^m = \frac{f_o^m b - a v^m}{b + v^m} \quad (3.6)$$

Inversely, we can calculate the velocity from the measured tension:

$$v^m = b \frac{f_o^m - f^m}{f^m + a}, \quad (3.7)$$

$$v_o^m = b \frac{f_o^m}{a}. \quad (3.8)$$

v_o^m is the maximum velocity of shortening and is experienced when there is no load on the muscle. The muscle exerts maximal force when it is isometric such that v^m is zero. The coefficients, a and b represents the effects of temperature and the type of the muscle (slow, fast or mixed fibers) respectively.

Combining the elements

By combine the effects of all the elements (ignoring the series element), the force generated is computing by adding the contribution from the parallel and contractile elements:

$$f^m = f_{PE} + f_{CE}. \quad (3.9)$$

3.2.2 Muscle line of action

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. This introduces some possible inaccuracies. One is that real insertion and origin attachment regions are areas, not points. This is adequate 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 available on the distribution of muscle effort within 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.

In light of the lack of experimental justification for a more complex model, we adopted the piecewise linear model. 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 the production software system called *Maya*. Then we estimated the origins and insertions to be the centers of the muscle or tendon attachments to the bones (see Figure 3.4).

In Chapters 4 and 5 we will discuss how the force-length relationship and the piecewise line segment approach are used to compute the muscles' force magnitudes and directions.

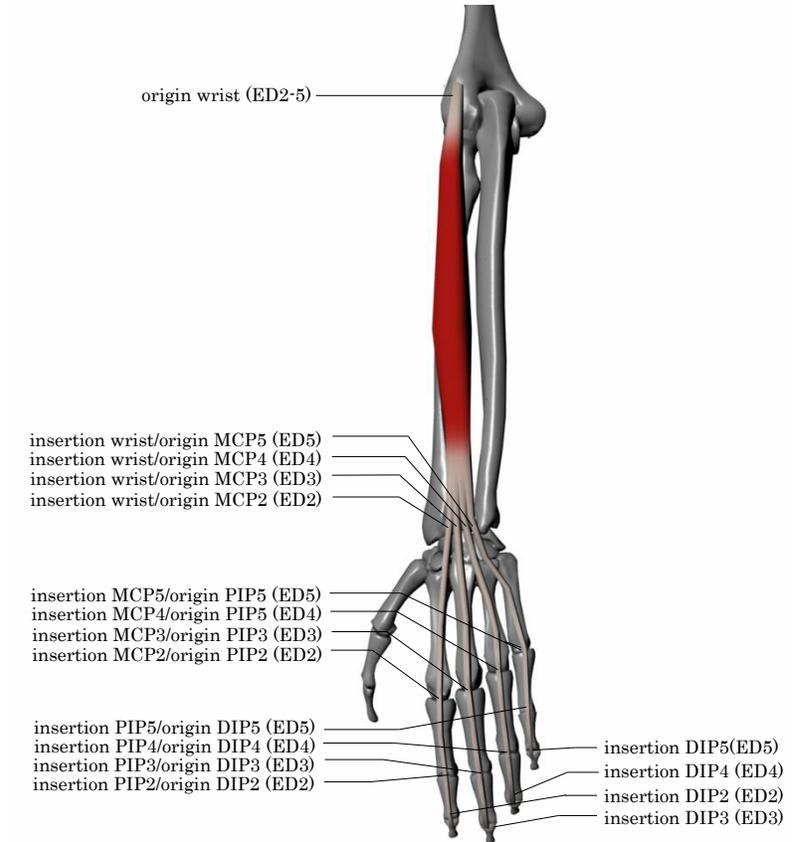


Figure 3.4: Extensor Digitorum forking into multiple tendons spanning multiple joints.

Table 3.1: Functions of Extrinsic Muscles

Muscle of Anterior Compartment	Main Actions
Pronator teres (PT)	Pronates forearm and flexes it
Flexor carpi radialis (FCR)	Flexes hand and abducts it
Palmaris longus (PL)	Flexes wrist and adducts it
Flexor carpi ulnaris	Flexes wrist and adducts it
Flexor digitorum superficialis (FDS)	Flexes middle phalanges of medial four digits; acting more strongly, it flexes proximal phalanges and wrist
Flexor digitorum profundus (FDP)	Flexes distal phalanges of medial four digits; assists with flexion of wrist
Flexor pollicis longus (FPL)	Pronates forearm
Pronator quadratus (PQ)	Pronate forearm
Muscle of Posterior Compartment	Main Actions
Brachioradialis	flexes forearm
Extensor carpi radialis longus (ECRL)	Extend and abduct wrist
Extensor carpi radialis brevis (ECRB)	Extend and abduct wrist
Extensor digitorum (ED)	Extends medial four digits at metacarpophalangeal joints; extends wrist
Extensor digiti minimi (EDM)	Extends fifth digit at metacarpophalangeal joints; and interphalangeal joints
Extensor carpi ulnaris (ECU)	Extends and adducts hand at wrist
Abductor pollicis longus (APL)	Abduct thumb and extends it at carpometacarpal joint
Extensor pollicis brevis (EPB)	Extends proximal phalanx of thumb at carpometacarpal joint
Extensor pollicis longus (EPL)	Extends distal phalanx of thumb at metacarpophalangeal and interphalangeal joints
Extensor indicis (EI)	Extends second digit and helps to extend wrist

Table 3.2: Functions of Intrinsic Muscles

Muscle of Hand	Main Actions
Abductor pollicis brevis (APB)	Abducts thumb and help oppose it
Flexor pollicis brevis (FPB)	Flexes thumb
Opponens pollicis (OP)	Opposes thumb toward center of palm and rotates it medially
Adductor pollicis (AP)	Adducts thumb toward middle digits
Abductor digiti minimi (ADM)	Abducts digits 5
Flexor digiti minimi brevis (FDMB)	Flexes proximal phalanx of digit 5
Opponens digiti minimi (ODM)	Draws fifth metacarpal bone anteriorly and rotates it, bringing digit 5 into opposition with thumb
Lumbricals 1 (LUMB I)	Flex digit 2 at metacarpophalangeal joint and extend interphalangeal joint
Lumbricals 2 (LUMB II)	Flex digit 3 at metacarpophalangeal joint and extend interphalangeal joint
Lumbricals 3 (LUMB III)	Flex digit 4 at metacarpophalangeal joint and extend interphalangeal joint
Lumbricals 4 (LUMB IV)	Flex digit 5 at metacarpophalangeal joint and extend interphalangeal joint
Dorsal interossei 1 (DI I)	Adducts digit 2 and assist Lumbrical 2
Dorsal interossei 2 (DI II)	Adducts digit 3 and assist Lumbrical 3
Dorsal interossei 3 (DI III)	Adducts digit 3 and assist Lumbrical 3
Dorsal interossei 4 (DI IV)	Adducts digit 4 and assist Lumbrical 4
Palmar interossei 1 (PI I)	Adducts digits 2 and assist Lumbrical 2
Palmar interossei 2 (PI II)	Adducts digits 4 and assist Lumbrical 4
Palmar interossei 3 (PI III)	Adducts digits 5 and assist Lumbrical 5

Chapter 4

Anatomical 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. In this chapter, we will elaborate on these components.

4.1 Models of Joints

There are 16 joints in our hand model. Joints are modeled as hinges that are capable of rotation around the principal axes of its right handed co-ordinate systems. In the local co-ordinate system of a joint, the x -axis is always aligned with the bone that is linking it to its child joint. We define the the x - y plane as the *sagittal* plane, in which a rotation around the z -axis is pure flexion and extension. The x - z plane is the *coronal* plane, where rotations around the y -axis produce pure abduction (moving away middle finger) and adduction (moving toward middle

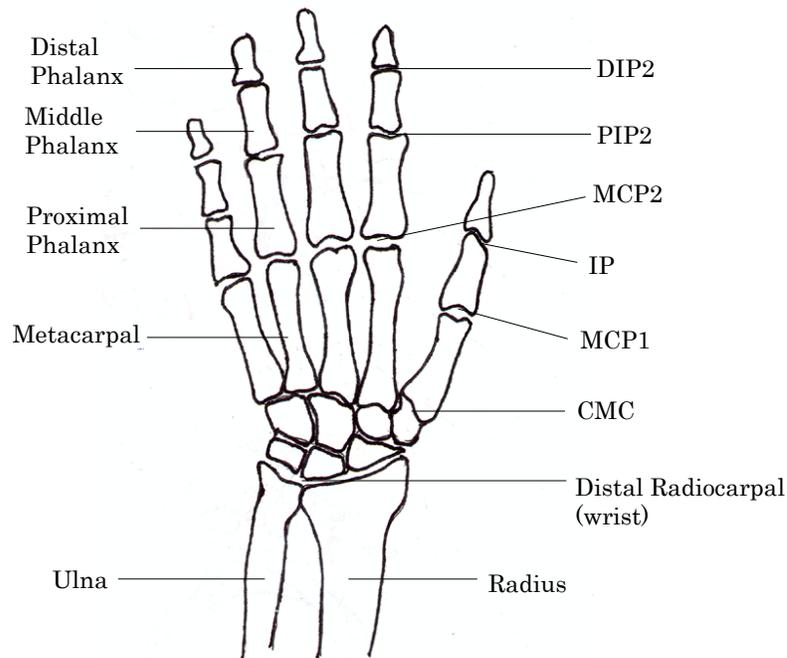


Figure 4.1: Hand joint nomenclature

finger). The y - z plane is the *transverse* plane, and rotating around the x -axis produces internal and external rotations, similar to twisting the bone.

Referring to Figure 4.1, 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 an 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 much more extensive discussion. Table A.1 shows the average ranges of motion for the joints.

4.2 Models of 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 4.1). The sizes and masses of each link were gathered from biomechanical sources [BH99, BY94] and measured from a 3D hand skeleton model, the data is tabulated in Appendix A.

4.3 Models of Muscles

There are in total 41 hand and forearm musculo-tendon units in our hand model. They are simplified as piecewise line segments having origins and insertion points instead of areas of attachments. To estimate these points, we manually fit digitized muscle fibers and tendon data to our 3D skeleton model using *Maya(tm)*. Then we estimate the origins and insertions to be the center of the muscle's or tendon's attachments to the bones. These points are tabulated in Table A.5. It should be noted that these values are relative to our hand skeleton, and thus not to be taken as valid anatomical data. The points are used for the purpose of calculating the direction of force exerted on the joint by a muscle or tendon crossing it. Although a more accurate model would model these points as areas, there is no experimental work available concerning the distribution of muscle effort inside the areas of muscle attachment. Despite this simplification, apart from the muscle origins located on forearm, the muscles insertions located on the wrists and fingers are very much thread-like.

To understand the distinction between muscle and musculo-tendon unit, consider the Flexor Digitorum Superficialis (FDS), Flexor Digitorum Profundus (FDP), and Extensor Digitorum (ED)(See Figure 3.4). 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 musculo-tendon units of each such muscle 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 musculo-tendon 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 musculo-tendons of FPS units can be contracted independently.

In chapter 3 we discussed how muscle exerts force. Adopting from [AHS03], we used the same quadratic curves that they have fitted to the two length-tension curves to calculate the magnitude of the force exerted by a muscle. The current fiber length is given by $l \in [0.6l_o, 1.6l_o]$, its resting length is given by l_o , and contraction value is given by $c \in [0,1]$. F_{max} is the maximum isometric force that can be exerted by a muscle. The contraction force term, corresponds to the contractile element in Hill's model, is:

$$F_c(l) = [1 - 4 \cdot (l/l_o - 1.1)^2] \cdot F_{max}. \quad (4.1)$$

The stretch force term, corresponds to the parallel element in Hill's model, is:

$$F_s(l) = \begin{cases} 2.77 \cdot (l/l_o - 1)^2 \cdot F_{max}, & l \geq l_o \\ 0, & l < l_o. \end{cases} \quad (4.2)$$

The total force is thus

$$F(l) = c \cdot F_c(l) + F_s(l). \quad (4.3)$$

F_{max} can be thought of as relative maximum strength of the muscle, and can be taken from Brand's paper [BBT81], in which a muscle's tension is compared to the total tension of all

studied muscles, and the ratio of strength from muscle to muscle within the same limb are quantified. These figures are useful for emulating different strengths of muscles in our physically-based hand model. We should emphasize that these values are relative and involve no numerical statement of tension. For example, a muscle with a tension fraction of 3 is capable of 50% more tension than a muscle with a tension fraction of 2. These values do allow us to easily capture the differences in the relative capabilities of the muscles.

Now that we can easily compute the magnitude of the force exerted, we address how these forces moves bones and joints. We called the point where the tendon attaches to the bone from the joint's center of rotation, the *lever arm*. Table A.4 tabulates the lever arms of the muscles as they cross the affected joints. These numbers are taken from graphs and tables from [BH99]. The total effect of a muscle at a joint is given by the torque, which is the cross product of the lever arm and the force. The magnitude of this cross product 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 the 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.

In Chapter 5 we will discuss how we use the force directions, and lever arms in the calculation of torques for simulation.

Chapter 5

Dynamic Simulation

In this chapter we will describe simulation of an unconstrained rigid-body system, and discuss techniques in integrating equations of motion. Finally, we explain the dynamics of our physically-based hand model, and the approach we took to perform simulation.

5.1 An Unconstrained Rigid-Body System

A rigid body is composed of particles, and its behaviour can be modelled using particles. In fact, most of the equations for a system of particles are usable in the dynamics of a rigid body. The main difference is of course that in the body, such that there is no migration of mass within a rigid body. As a result, the relative positions among the particles composing a rigid body do not change. This will further simplify the equations obtained from a system of particles.

In an unconstrained rigid-body system, the motion of the bodies are not constrained by contact forces that prevent inter-penetration. In a realistic physical system, non-penetration constraints should be enforced by computing appropriate contact forces between contacting bodies. However, computing these forces are demanding, as they involve collision detection, calculation

of colliding contact forces and resting contact forces. Thus, in our hand system, we used an unconstrained rigid-body system in order to keep the system simple enough for real-time simulation and the mathematics simple enough for the constraint-based approach where the gradient of the objective function needs to be symbolically derived. It should be noted that in our system there are limits placed on the rotation angles of the joints (Table A.1) and on the length of the muscles as they stretch or contract. This is to prevent the joints from rotating past their physical limits, and likewise, to prevent the muscles from stretching or contracting past their physical limits.

We now present a brief overview of simulating a physical system of n rigid bodies. Let \vec{S}_i be the state vector for a single rigid body i at time t , where \vec{x}_i is position, $\vec{\theta}_i$ is rotation, \vec{v}_i is velocity, and $\vec{\omega}_i$ is angular velocity. We write the state vector as

$$\vec{S}_i(t) = \begin{pmatrix} \vec{x}_i(t) \\ \vec{\theta}_i(t) \\ \vec{v}_i(t) \\ \vec{\omega}_i(t) \end{pmatrix} \quad (5.1)$$

For a system with n rigid bodies, the system's state vector is

$$\vec{S}(t) = \begin{pmatrix} \vec{S}_1(t) \\ \vdots \\ \vec{S}_n(t) \end{pmatrix} \quad (5.2)$$

To simulate the motion of rigid bodies, the forces and torques at any given time need to be computed. Let $\vec{F}(t)$ be the force, and $\vec{T}(t)$ be the torque acting on our system at time t . They are both vectors of length $3n$. Let \mathbf{M} be the $3n$ by $3n$ diagonal mass matrix whose off-diagonal

elements are zero, and whose diagonal elements are $[m_1, m_1, m_1, m_2, m_2, m_2, \dots, m_n, m_n, m_n]$. Similarly let \mathbf{I} be the inertia tensor matrix, with dimension of $3n$ by $3n$, with the diagonal elements are $[I_1, I_2, \dots, I_n]$, where I_i is the 3 by 3 inertia tensor for rigid body i . The equations of motion are as follows:

$$\frac{d}{dt} \vec{v}(t) = \mathbf{M}^{-1} \vec{F}(t), \quad (5.3)$$

$$\frac{d}{dt} \vec{\omega}(t) = \mathbf{I}^{-1} \vec{T}(t). \quad (5.4)$$

Using these equations, we can write first order differential equations for the state vector at time t :

$$\frac{d}{dt} \vec{S}(t) = \frac{d}{dt} \begin{pmatrix} \vec{x}(t) \\ \vec{\theta}(t) \\ \vec{v}(t) \\ \vec{\omega}(t) \end{pmatrix} = \begin{pmatrix} \vec{v}(t) \\ \vec{\omega}(t) \\ \mathbf{M}^{-1} \vec{F}(t) \\ \mathbf{I}^{-1} \vec{T}(t) \end{pmatrix} \quad (5.5)$$

To simulate the system, we need to solve a problem involving ordinary differential equations. The next section highlights some of the methods in accomplishing this.

5.2 Integration of Ordinary Differential Equations

Problems involving ordinary differential equations can be decomposed to a set of first-order differential equations. The following is a typical second-order equation of motion:

$$m \frac{d^2 x}{dt^2} + k \frac{dx}{dt} = F(t), \quad (5.6)$$

can be rewritten as two first order equations

$$v(t) = \frac{dx}{dt}, \quad (5.7)$$

$$\frac{d^2 x}{dt^2} = \frac{F(t) - kv(t)}{m}. \quad (5.8)$$

There are many methods of solving ordinary differential equations (ODE). They differ in their computational complexity and stability. For example, explicit euler integrator is the simplest. However, the time-step taken must be small or else the system will become unstable. In contrast, implicit euler integrator is more complex, but a larger time-step can be taken in simulation. In this section will discuss the following types of integrators:

- Explicit Euler method,
- Implicit Euler method,
- Runge-Kutta method.

A deeper treatment of this material can be found in a textbook on scientific computation, such as [PTVF02].

5.2.1 Explicit Euler Method

The formula for *Explicit Euler* method is

$$y_{n+1} = y_n + hf(y_n) \tag{5.9}$$

Note that the derivative information is only used at the beginning of that interval, and by power series expansion, it can be observed that it is only first order accurate.

$$y_{n+1} = y_n + hf(y_n) + \frac{h^2}{2}\dot{f}(y_n) + \dots \tag{5.10}$$

Explicit Euler method is very simple but not recommended for practical use, since it is not very accurate, and unstable.

5.2.2 Implicit Euler Method

The formula for *Implicit Euler* method is

$$y_{n+1} = y_n + hf(y_{n+1}) \quad (5.11)$$

This requires the function be evaluated at a future time step, thus, $f(y_{n+1})$ is approximated as

$$f(y_{n+1}) = f(y_n) + \frac{\partial f}{\partial y}|_{y_n}(y_{n+1} - y_n) \quad (5.12)$$

Implicit euler method is also only first order accurate but it is stable, and is thus a more suitable method for ODE integration.

5.2.3 Runge-Kutta Method

Runge-Kutta or midpoint methods propagate a solution over an interval by combining the derivative information from several Explicit Euler steps. Instead of taking one complete step, we take steps to the midpoint of the interval as illustrated below

$$k_1 = hf(y_n) \quad (5.13)$$

$$k_2 = hf\left(y_n + \frac{1}{2}k_1\right) \quad (5.14)$$

$$y_{n+1} = y_n + k_2 \quad (5.15)$$

This method is second order accurate, but it is not fast unless the computation of derivatives happens to be cheap. More steps can be used, as in the fourth-order Runge-Kutta method. The step-size can also be adaptive for gains in efficiency so that smaller steps are used for areas where the function changes rapidly, and larger steps only for areas where the function is fairly flat.

5.3 Dynamic Simulation of Hand Model

In this section we explain 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 . We 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

$$\vec{S}_i(t) = \begin{pmatrix} \vec{\theta}_i(t) \\ \vec{\omega}_i(t) \end{pmatrix}. \quad (5.16)$$

To compute the total torque at joint i , we need to sum up all the contribution of 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 also introduce a set of weights, $\vec{\alpha}=[\alpha_1, \alpha_2, \dots, \alpha_M]$, to control the amount of contribution to the total tension that a muscle can exert at time t . These weights have a value between zero and one, and model the effect of muscle fatigue. For instance, a muscle with a weight value of zero cannot exert any tension at all, whereas a muscle with a weight value of one exerts tension as in normal operation. If an arbitrary number of m muscles crosses joint i , then its total torque, \vec{T}_i at time t is given by

$$\vec{T}_i(t) = \sum_{j=1}^m \vec{r}_{i,j} \times \alpha_j(t) \vec{F}_{i,j}(t). \quad (5.17)$$

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

$$\vec{F}_{i,j}(t) = [c_j F_{c,j}(l_j) + F_{s,j}(l_j)] \hat{F}_{i,j}, \quad (5.18)$$

where

$$\hat{F}_{i,j} = \frac{i_{i,j} - o_{i,j}}{\|i_{i,j} - o_{i,j}\|}. \quad (5.19)$$

Let \mathbf{I}_i be a 3×3 inertia tensor for joint i , and k_μ be the coefficient of friction. The second-order ordinary differential equation of motion for joint i is given by

$$\vec{T}_i(t) = \mathbf{I}_i(t) \frac{d}{dt} \vec{\omega}_i(t) + k_\mu \vec{\omega}_i(t). \quad (5.20)$$

and can be rewritten as two first-order ordinary differential equations:

$$\begin{aligned}\frac{d}{dt}\vec{\theta}_i(t) &= \vec{\omega}_i(t), \\ \frac{d}{dt}\vec{\omega}_i(t) &= \mathbf{I}_i^{-1}(t) \left(\vec{T}_i(t) - k_\mu \vec{\omega}_i(t) \right) = f(\vec{\omega}_i(t)).\end{aligned}\tag{5.21}$$

Now we extend to a system of n joints, where $\vec{\theta}$, $\vec{\omega}$, \vec{T} are $3n \times 1$ vectors, \mathbf{I} is a $3n \times 3n$ matrix.

Using the *Implicit Euler* method for integration,

$$\begin{aligned}\vec{\omega}_{t+1} &= \vec{\omega}_t + \Delta t f(\vec{\omega}_{t+1}) \\ &= \vec{\omega}_t + \Delta t \left(f(\vec{\omega}_t) + \frac{\partial f}{\partial \vec{\omega}} \Big|_{\vec{\omega}_t} (\vec{\omega}_{t+1} - \vec{\omega}_t) \right).\end{aligned}\tag{5.22}$$

Expressing $\vec{\omega}_{t+1} - \vec{\omega}_t$ as $\Delta \vec{\omega}$,

$$\begin{aligned}\Delta \vec{\omega} &= \Delta t f(\vec{\omega}_t) + \Delta t \frac{\partial f}{\partial \vec{\omega}} \Big|_{\vec{\omega}_t} \Delta \vec{\omega} \\ f(\vec{\omega}_t) &= \Delta \vec{\omega} \left(\frac{1}{\Delta t} - \frac{\partial f}{\partial \vec{\omega}} \Big|_{\vec{\omega}_t} \right).\end{aligned}\tag{5.23}$$

Substituting Equation 5.21 for $f(\vec{\omega}_t)$ and differentiating with respect to $\vec{\omega}$ to obtain $\frac{\partial f}{\partial \vec{\omega}} \Big|_{\vec{\omega}_t} = -k_\mu \mathbf{I}_t^{-1}$,

$$\begin{aligned}\mathbf{I}_t^{-1} \left(\vec{T}_t - k_\mu \vec{\omega}_t \right) &= \Delta \vec{\omega} \left(\frac{1}{\Delta t} + k_\mu \mathbf{I}_t^{-1} \right) \\ \vec{T}_t - k_\mu \vec{\omega}_t &= \Delta \vec{\omega} \left(\frac{\mathbf{I}_t}{\Delta t} + k_\mu \right).\end{aligned}\tag{5.24}$$

The system of equations are in the form of $\mathbf{A}\vec{x}=\vec{b}$, where

$$\begin{aligned}\vec{x} &= \Delta \vec{\omega}, \\ \mathbf{A} &= \frac{\mathbf{I}_t}{\Delta t} + k_\mu, \\ \vec{b} &= \vec{T}_t - k_\mu \vec{\omega}_t.\end{aligned}\tag{5.25}$$

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:

$$\vec{\omega}_{t+1} = \vec{\omega}_t + \Delta\vec{\omega}, \quad (5.26)$$

$$\Delta\vec{\theta} = \vec{\omega}_{t+1}\Delta t, \quad (5.27)$$

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \Delta\vec{\theta}. \quad (5.28)$$

We also compute the new length of muscle j as

$$l_j = l_j - \sum_{i=1}^n \Delta\vec{\theta} \cdot \vec{r}_{i,j} \quad (5.29)$$

where,

$$l_j = \begin{cases} 1.5 \cdot l_o & , l \geq 1.5 \cdot l_o \\ 0.5 \cdot l_o & , l \leq 0.5 \cdot l_o, \end{cases} \quad (5.30)$$

to clamp the length of the muscle is its upper and lower limits of $1.5 \cdot l_o$ and $0.5 \cdot l_o$ respectively.

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 one or more 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. For the wrist, we assume a constant inertia tensor and compute it once prior to simulation. The tensor is approximated by distributing points on the surface of the joint, calculating moments of inertia in a pointwise fashion and accumulating the result. The mathematics for inertia tensor computation is in Appendix B.

Chapter 6

Control: Constraint-based approach

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 minimizes the scalar function. In this chapter we describe the control of our system using energy constraints. First, an explanation of the concept behind energy constraints is presented, followed by the formulation of the constraints functions for our system. Subsequently, the process of minimizing these function is described.

6.1 Energy Constraints of Hand Model

In a forward physical simulation, the model predicts the motion given a set of forces as inputs. The forward model's 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 $\vec{\omega}_i$:

$$\vec{y} = f(\vec{c}) = \begin{pmatrix} \vec{\theta}_1 \\ \vec{\omega}_1 \\ \vdots \\ \vec{\theta}_n \\ \vec{\omega}_n \end{pmatrix}, \quad (6.1)$$

where the contraction values for m muscles are $\vec{c} = (c_1, c_2, \dots, c_m)$, 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 $\vec{t} = (t_1, t_2, \dots, t_{6n})$ over n joints. We formulate the *energy function* or the *objective function* as

$$E(f(\vec{c})) = w_s \frac{1}{2} E_s(\vec{y}) + w_c \frac{1}{2} E_c(\vec{c}) + w_m \frac{1}{2} E_m(\vec{c}). \quad (6.2)$$

This is a weighted sum with scalar weights w_s , w_c , and w_m of the terms E_s , E_c , and E_m which respectively evaluate the motion, the controller, and the number of muscles contracted. 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. Finally, minimizing the number of muscles contracted using E_m prefers a solution that uses the fewest number of muscles. Mathematically, these energy function terms are expressed as

$$E_s(\vec{y}) = \|\vec{t} - f(\vec{c})\|_2^2, \quad (6.3)$$

$$E_c(\vec{c}) = \frac{1}{2} \|\vec{c}\|_2^2, \quad (6.4)$$

$$E_m(\vec{c}) = \frac{\|U(\vec{c} - c_o)\|_1}{m}, \quad (6.5)$$

where $\|\cdot\|_2$ denotes 2-norm of a vector, and $\|\cdot\|_1$ denotes the 1-norm. $U(c - c_o)$ denotes a unit step function with c_o chosen to be close to zero:

$$U(t - t_o) = \begin{cases} 1 & t \geq c_o, \\ 0 & t < c_o. \end{cases} \quad (6.6)$$

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. E_m minimizes the number of muscles contracted by counting all the contraction values greater than c_o .

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 all the goals

$$E(\vec{c}) = \sum_i w_i E_i. \quad (6.7)$$

6.2 Minimization of Function

The constraint-based approach poses an optimization problem such that given a function with one or more independent parameters, we want to find values for those parameters where f takes on a minimum or maximum value. There are various plausible optimization algorithms. Some only involve evaluations of the function, and others also require the evaluations of the the function's partial derivatives with respect to all parameters. In general, the cost of optimization involves evaluating f and perhaps also its gradient multiple times. Algorithms that use derivative information are more powerful but more computationally intensive.

Recall from Eq. 6.1 that the forward dynamics operation f is a nonlinear function of m muscle contraction values that are all in $[0,1]$. L-BFGS-B, a limited memory algorithm for solving

large nonlinear constrained or unconstrained optimization problem [?] is used for optimization. At each iteration, the optimizer requires a scalar value of the function evaluated at the current point in parameter space, and its vector gradient. For more details regarding the algorithm, refer to [BLN95]. A thorough description of the calculation of the gradient is found in Appendix D.

6.3 The Constrained Simulation Loop

The simulation of the constrained system is similar to the unconstrained case. At each discrete time-step, evaluations of the function are performed by solving the ordinary differential equations as in the unconstrained case, with the addition of derivative evaluations. There will be multiples of these evaluations per time-step until convergence of the optimization algorithm is reached.

A routine problem in optimization is that of finding a local optimum that is globally suboptimal. A global extremum optimizes the function. A local extremum is the highest or lowest value of the function within an interval. The quality of the result of the optimization algorithm is dependent on the initialization of the global parameters. If the initial guess is good, then convergence is quick, and the value of the energy function is small enough, indicating that the goals are satisfied. However, if the initial guess is bad, that the output of the algorithm is a poor local minimum, and the value of the energy function may be too large. To avoid bad initialization, we first define a threshold on the value of the energy function, *energyThreshold*, and define the maximum number of trials, *maxTrial*. Then we repeat the optimization procedure until the value of energy function is less than *errorThreshold* or the number of trials exceeds *maxTrial*. At each iteration we use the local minimum found in the previous iteration as initial parameters. The set of parameters that produced the lowest energy function value is kept. To summarize, the steps in the simulation loop is shown:

```
While (energy  $\geq$  energyThreshold or trial  $\leq$  maxTrial) {  
  1. Initialize parameters,  $\vec{c}$   
  2. Optimize parameters:  
    (a) Evaluate energy function  
    (b) Evaluate gradient energy function  
    (c) Perform a LBFGS approximation to the Hessian Matrix  
    (d) Compute search direction  
    (e) Perform line search in the direction computed  
    (f) Repeat steps (a) to (e) until one of the LBFGS stopping conditions is reached  
    (g) Return energy'  
  3. If (energy'  $\leq$  energy) energy = energy', and save parameters,  $\vec{c}$   
}
```

Once a stopping conditions is met, the system is simulated with the saved parameters, and we proceed to next time-step in the animation. At the end of the animation, the result would be a set of parameters key-framed at each time-step.

More than twenty tendons cross the wrist joint. This makes it difficult to search efficiently 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 similar to the procedure we just described; however, we only include as degrees of freedom a restricted set of seven muscles that only affect the wrist joint. Specifically, these muscles are FCR, PL, FCU, ECRL, ECRB, ECU, and APL, which act as wrist stabilizers. Finally, we combine the resulting seven joint contraction values with the previously solved ones and forward simulate the system. The way that muscles of the hand actually work together is still unknown; that is, it is not known if our nervous system

activates the finger muscles to induce the desired motion and then activates the wrist muscles to stabilize the wrist. It is thus possible that our method of isolating the wrist joint may not be biologically valid and future work will be needed to validate this approach.

6.4 Exploring the Solution Space

Computing an inverse dynamics solution is a challenging problem due to the redundancy of muscles that allow different set of muscles to be used to give similar motion. We used various clinically motivated heuristics for repetitive strain injury (R.S.I.) diagnosis to explore the solution space by simulating muscle fatigue by means of setting musculo-tendon unit weights, and solving once again for an alternative solution.

User-controlled. The weight attached to musculotendon units can be manipulated by the user to selectively disable musculo-tendon units from the solution by assigning weights of zero, or to lessen their c with smaller weights. Figure 6.1(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 musculo-tendon units, and with only the active musculo-tendon units plus a small subset enabled, the latter approach gives a more accurate match to the target motion.

Top-usage. This approach makes use of previous solutions to find the musculo-tendon units that are mostly used by summing up their contraction values throughout the entire animation sequence. Thus, the primary muscles will be assigned a weakened weight in the inverse simulation. Figure 6.1(d) shows the results of this approach.

Threshold. This approach also makes use of previous solutions to find fatigued muscles by searching for musculo-tendon units with their contractions or lengths exceeding the set thresholds. Thus, musculo-tendon units that are stretched or pulled or contracted past their thresholds

are assigned a weakened weight. Figure 6.1(e) shows the results of this approach.

Random Combination. In this approach we randomly select an user-defined number of musculo-tendon units and assign a weakened weight to them. Figure 6.1(f) shows the results of this approach.

Figure 6.1 shows the results using the various techniques for simulating muscle fatigue outlined above. The results are worse compared to the best fit solution, as would be expected.

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, as we will describe shortly, because it is possible to create different affinities for muscle activations, our control method is able to search for quite different solutions in the case, for example, of muscle fatigue, pathology or atrophy. Third, because our physical model captures captures crosstalk across muscles and joints, the control solutions will naturally exhibit this effect. Our model reflects the fact that the human hand is not composed of a set of fully independently adjustable actuators.

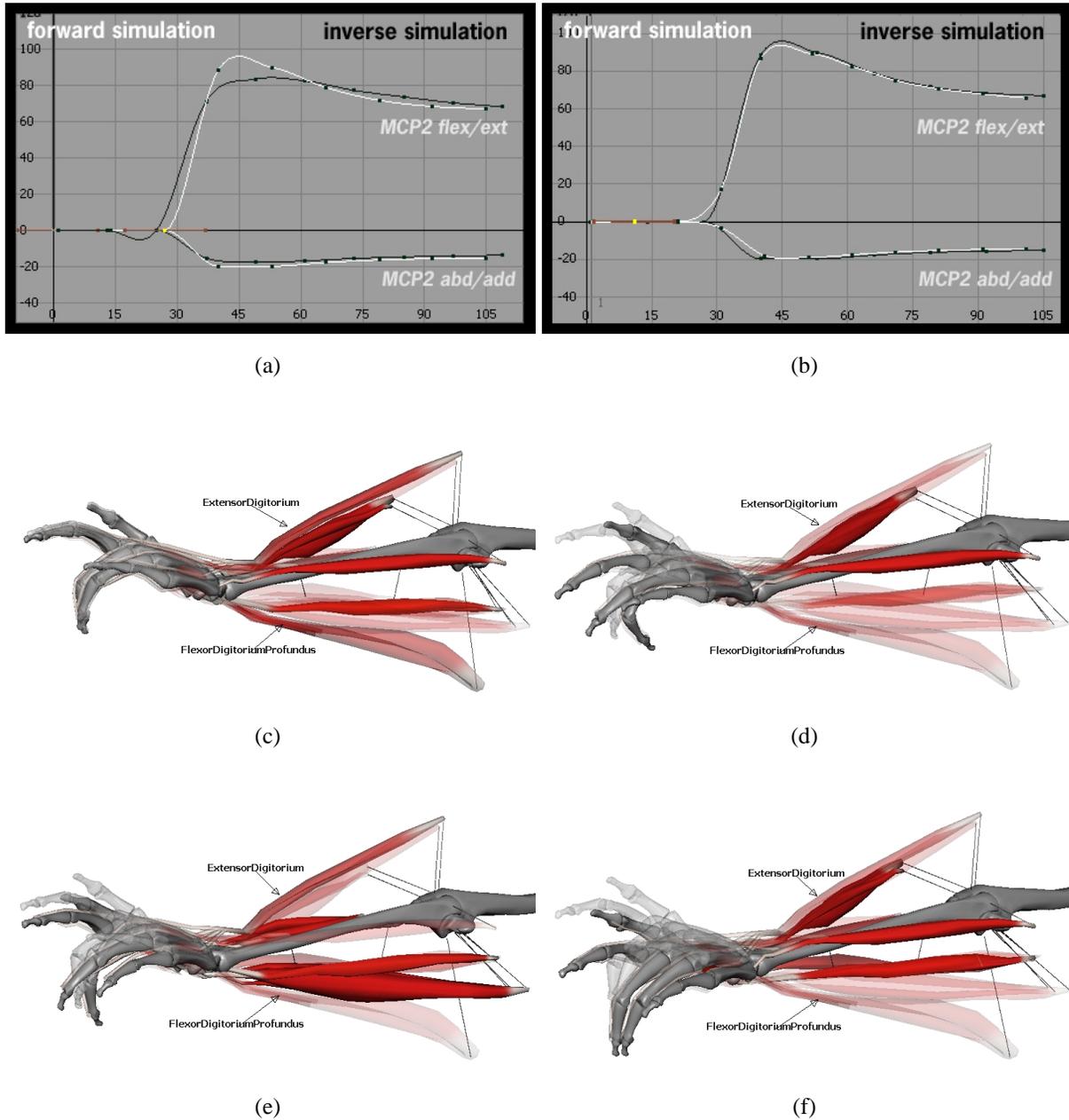


Figure 6.1: Exploring inverse solution space: (a) Animation curve of inverse solution for MCP2 using all muscles, (b) Animation curve of inverse solution for MCP2 using a superset of active muscles, (c) Best Fit Solution (d) Solution without primary muscles, (e) Solution without fatigued muscles, (f) Randomly weighted solution.

Chapter 7

Visualization

This chapter describes the visualization techniques used to present the intricate anatomical data of the hand in a comprehensible manner.

7.1 Visualization Techniques

Due to the complexity of the anatomical information of the hand, it is a challenging task to present it in a form that is visually pleasing, and easily understandable. We have implemented a variety of visualization techniques to depict hand animation. Our goal through these visualization techniques is to present the architecture and motion of the hand to clinicians and animators in ways that illuminate hand function. Our techniques are inspired from anatomical atlas [AL99], plastination techniques (Figure 7.3a), and existing techniques of visualizing volumetric data [MTB03].

7.1.1 Transparency

We demonstrate muscle activity through the use of opacity (see Figure 7.1.1) producing less obscured view of actively contracting muscles. The transparency of the muscle-tendon units changes linearly with its contraction value, such that they are semi-transparent when there is no contraction and become fully opaque when fully contracted.

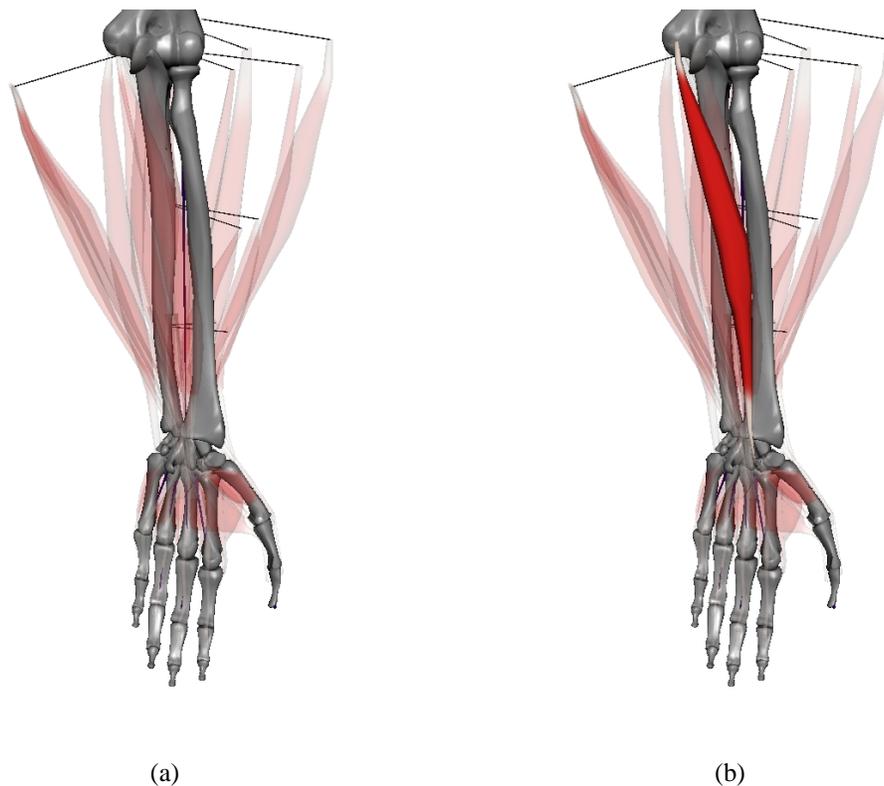


Figure 7.1: Muscle contraction controlling transparency: (a) No muscle activated (b) One muscle activated

7.1.2 Muscle Deformation

Muscles also bulge under tension. See Figure 7.1.2 which is visually easier to perceive than variations in lengths, providing compelling feedback on muscle contraction. To accom-

plish this, for each musculo-tendon unit, two blend shapes are created for the contracted and stretched appearance. The musculo-tendon unit's length attribute controls the factor of blending between these shapes.

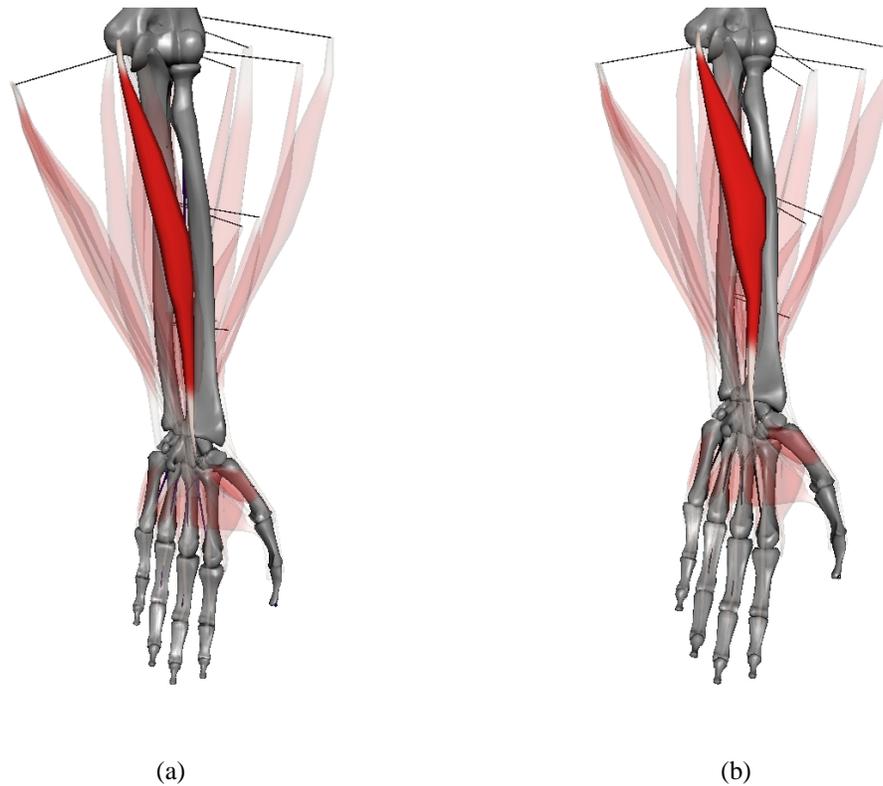


Figure 7.2: Muscles bulge under tension: (a) Muscle at rest length (b) Muscle at contracted length.

7.1.3 Spreading

The complex and compacted muscle architecture of the hand makes it difficult to visualize the deep structures. Adopting from [AL99], we categorized the forearm musculo-tendon units into 6 groups. Referring to Figure 3.1, three groups for the anterior aspect and three groups for the posterior aspect of the forearm, and those three group are divided based on depth. Each group can be “spread” by positioning the associated controller objects. The visibility of each

group can also be changed by controlling the *visibility* attribute of the controller objects. The “spreading” of muscle is inspired by Gunther Von Hagen’s plastination works [Hag]. His plastination technique on real human body specimens, including whole bodies, healthy and unhealthy organs, body parts and slices, reveals significant insights about human anatomy, physiology and health. To display the anatomical structure of the human hand, muscles can be peeled, individually or as a group, from their attachment. To keep the attachment information in context as the muscle is being spread, we attach a cylindrical polygon from the muscle’s end to its attachment point. Figure 7.3 shows the exploration of hand musculoskeletal architecture by spreading out muscles, with the selected muscle highlighted and annotated.

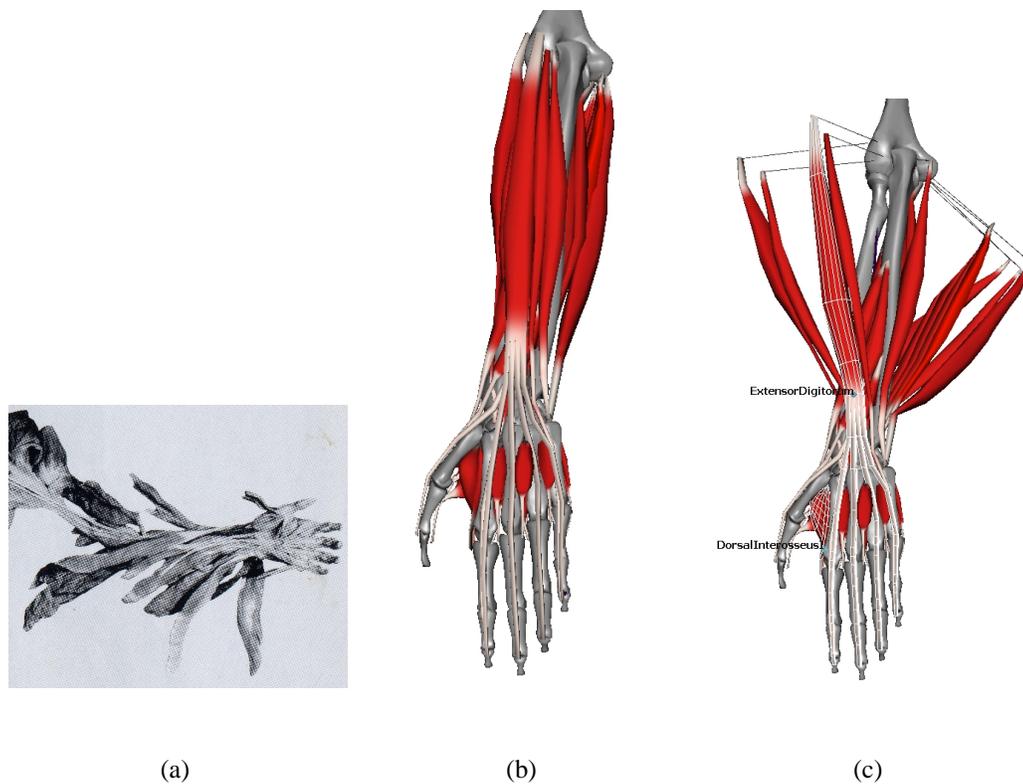


Figure 7.3: Exploring hand musculoskeletal architecture by fanning out muscles, highlighting and annotating selected musculotendons. Figure (a) shows the motivational plastinated hand

©Gunther Von Hagen

7.1.4 Ghosting

We use the technique of ghosting to present the error in the inverse solution, see Figure 7.4. A ghosted hand (a semi-transparent hand), moving with respect to the targeted motion, is overlaid on the hand with the inverse solution. The animations of these hands are played simultaneously to observe the error between the inverse solution and the desired hand configuration. The ghosted hand's visibility can be switched on or off according to user's preference.

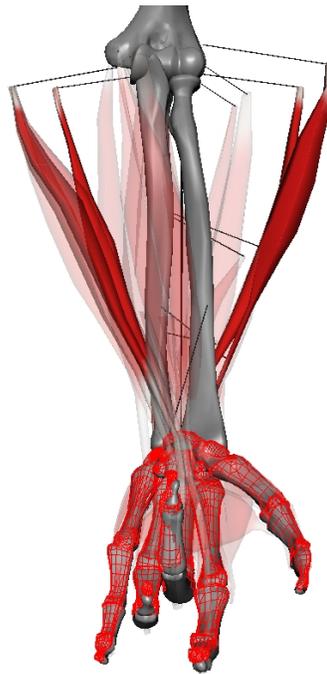


Figure 7.4: Ghosted skeleton, highlighted in red, indicates simulation error

Chapter 8

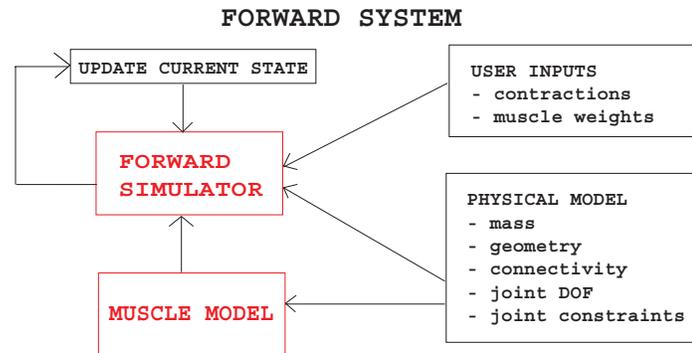
Implementation and Results

This chapter presents the implementation details of the system we call *Helping Hand* and its clinical and animation applications.

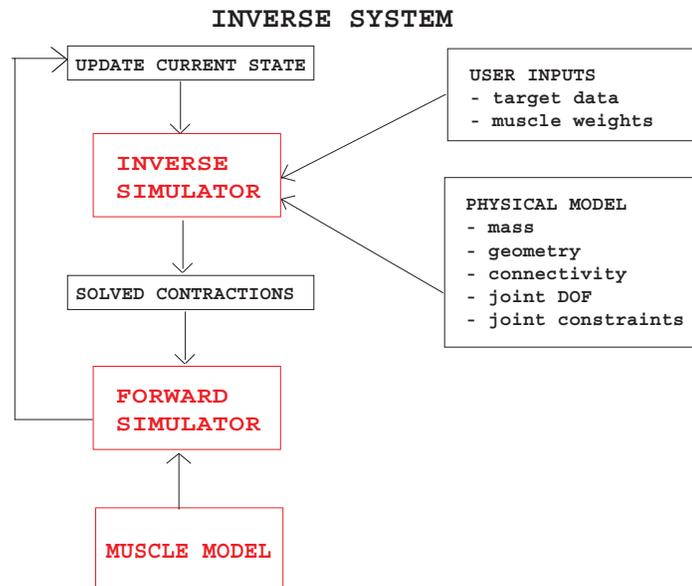
8.1 Modularity of the System

Helping Hand is composed of three modules: the forward simulator, the inverse simulator, and the muscle's force model as shown in Figure 8.1. Figure 8.2 shows a screenshot of the Helping Hand scene implemented in *Maya's* animation platform.

The forward simulator takes the muscle activation values and weights as input, which are defined by the user by keyframing the musculo-tendon units' attributes using the *Maya* interface. Figure 8.3 shows the interface for keyframing musculo-tendon units' contractions. It performs implicit integration of the equation of motions (described in Chapter 5) for each user defined time step. Typically, by using a time step no larger than 0.5 seconds, or equivalently 12 frame step, for a frame rate of 24 frames per sec, allows for stable integration. Using a time step too large, will yield large fluctuations in rotations and angular velocities. The forward simulator



(a)



(b)

Figure 8.1: System architecture of Helping Hand: (a) the forward simulator, (b) the inverse simulator

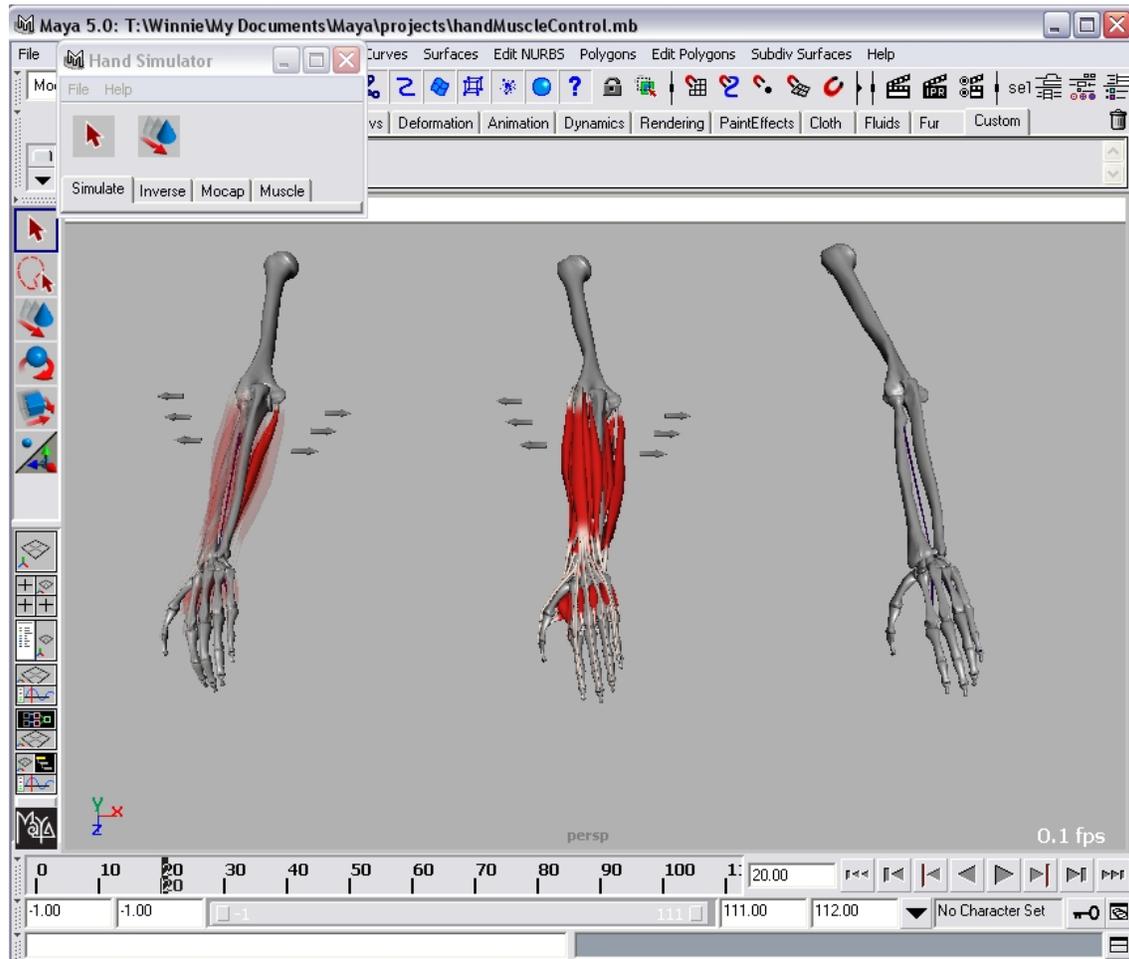


Figure 8.2: Screenshot of the Helping Hand scene implemented in *Maya*'s animation platform. To the left is the hand for forward simulation, to the middle is the hand for inverse simulation, and to the right is the hand for loading in target motion data

produces finger configurations each frame step, and the resulting animation can be playback by advancing the time slider in the *Maya* interface.

The inverse simulator takes finger configurations as input, which are joint rotations, and 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 time step. The time 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.

8.2 Application and Results

8.2.1 Clinical Application

Helping Hand provides anatomists with tools for visualizing the anatomical structure of hand muscles and exploring muscles' functions. Using the forward simulation component, they can manipulate activation values to obtain the resulting motion for contracting particular muscles. Figure 8.4 depicts the result from forward dynamics simulation with FDS2 having an activation of 1 throughout 100 frames of animation. Figure 8.5 depicts the result of the same simulation, now with FDS2 having a weakened muscle weight of 0.2.

With the inverse simulation component, anatomists can examine which muscles are utilized to produce unconstrained hand motion. The redundancy of muscles allows different sets of mus-

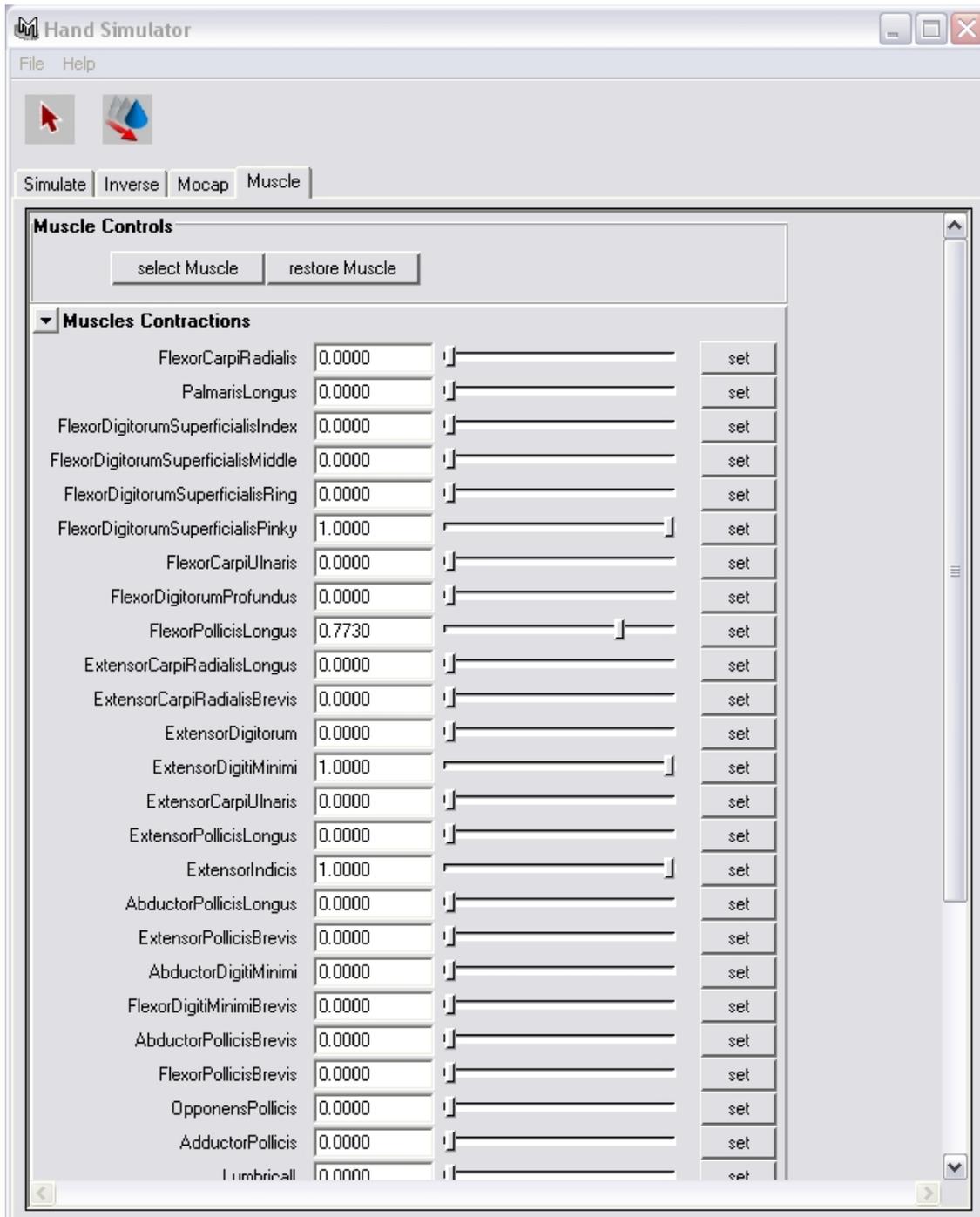


Figure 8.3: Interface for keyframing musculo-tendon units' contractions.

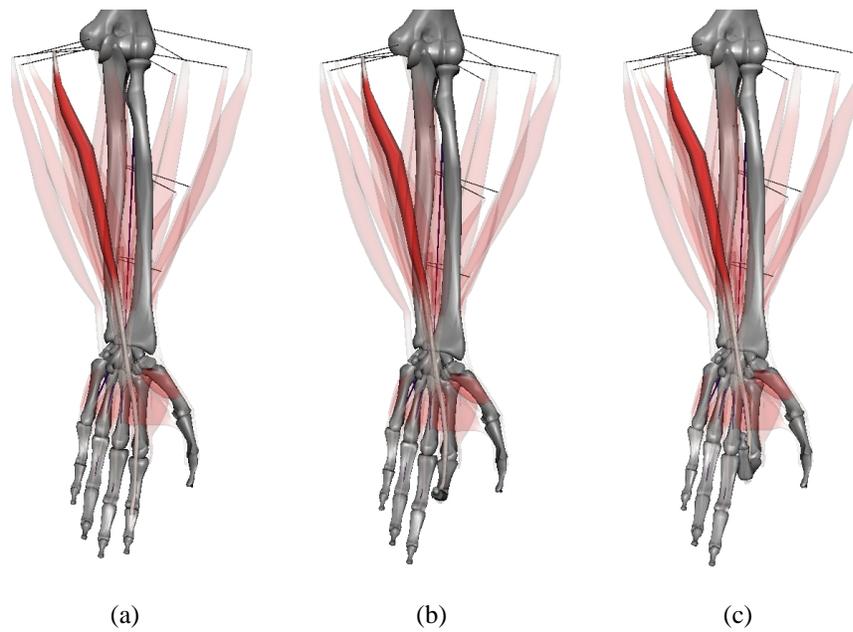


Figure 8.4: Forward simulation sequence due to contraction of non-weakened musculo-tendon units FDS2.

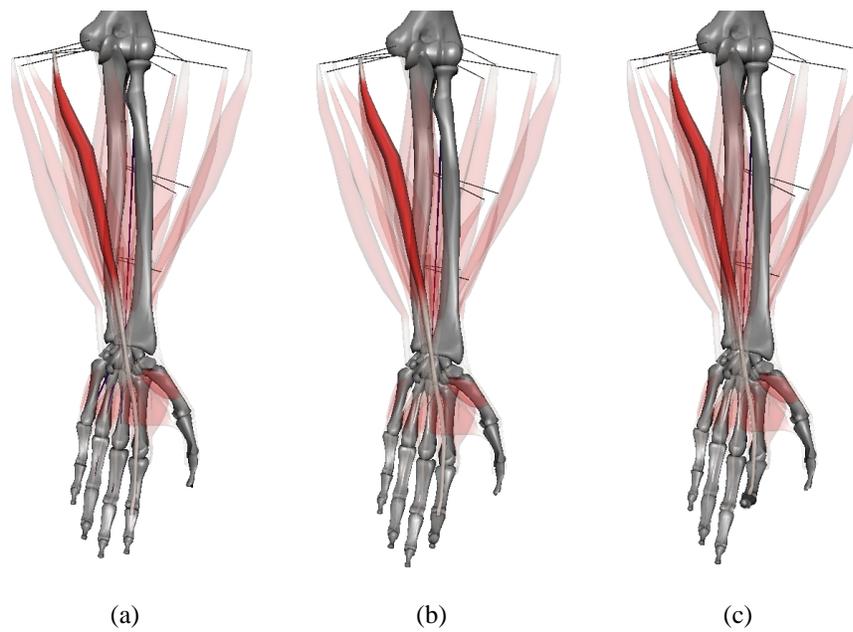


Figure 8.5: Forward simulation sequence due to contraction of weakened musculo-tendon units FDS2.

cles 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, which may be the result, for example, of either muscle failure or a psychological issue. Thus, anatomists can manipulate muscle weights using various clinically motivated heuristics for R.S.I diagnosis for exploring the solution space, as described in Chapter 6, to cause the system to optimize to a solution that favors the use of muscles with higher weights.

To validate our model against realistic unconstrained hand motion, we recorded a male and female subject performing various gestures as input to our inverse simulation component. This data is obtained with a optical motion capture system, by tracking the position of circular markers, placed at the 16 joints. Using this data, we can validate and, in the future, calibrate our hand model by observing the errors in the inverse solution. Figure 8.6 shows the result of the inverse simulation fitted to hand motion capture data. The errors for the digits were quite small, while, the error for the wrist was the largest, since wrist stabilization is an challenging biomechanical problem.

8.2.2 Animation Application

Helping Hand can also be used to validate and improve the quality 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 nonrealistic, 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

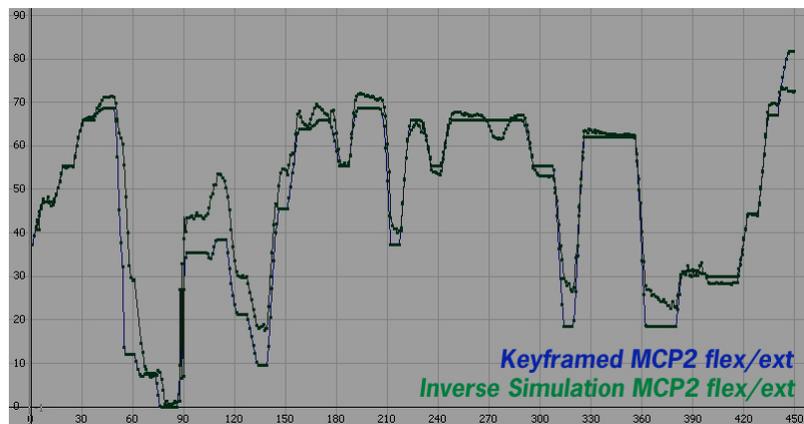


Figure 8.6: *Clinical application*: (a) and (b) show inverse simulation solutions to motion drills. *Animation application*: (c)-(e) show an inverse solution to the motion capture of the letters "A", "S", "L" performed in A.S.L. A ghosted hand shows the fitting error.

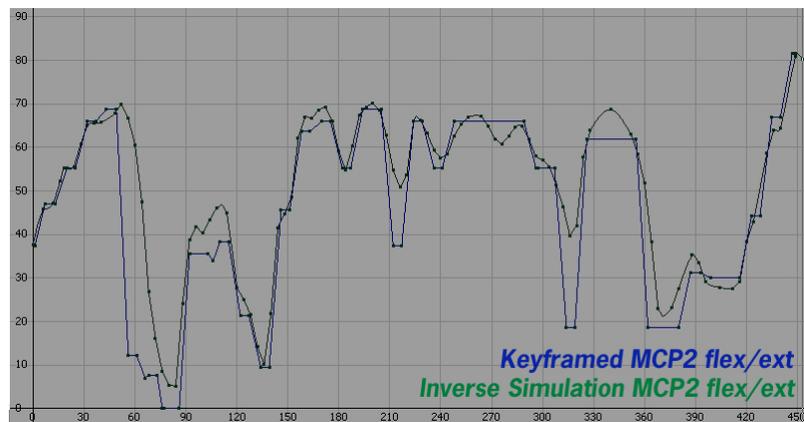
both approaches are shown in Figure 8.7 for comparison.



(a)



(b)



(c)

Figure 8.7: Simulating a hand keyframed using motion capture based IK postures. (a) Hand-drix animation, (b) Simulating a hand keyframed using motion capture based IK postures, (c) Simulation matching only the keyframed postures

Chapter 9

Conclusion

This work is an exploration of the human hand in an anatomical context. We have constructed a complete anatomically based hand model incorporating the hand and forearm's muscles, tendons, and bones. A time-varying neural control signal is used to stimulate a musculo-tendon unit to produce tension according to the Hill three-element model, which allows for a parameterizable representation to capture the individual characteristics of muscles and tendons. The system takes as input a set of muscle activations and output the resulting finger joint configuration by (forward) simulation. In addition, we present a solver that can calculate the muscle activations necessary to achieve a given pose or motion. The strength of a muscle is also parameterizable to model muscle fatigue, injury or atrophy, thereby yielding different control solutions that favor healthy muscles. Since the inverse problem can have many (or no) solutions, we demonstrate how the space of possible solutions can be explored by filtering this space using clinically motivated heuristics for R.S.I. diagnosis. The resulting motion from both forward and inverse simulation accounts for joint interdependency as a product of coupling nerve stimuli, muscle and tendon configurations. The system can take kinematic pose data such as motion capture or keyframed data as input, and it can predict animation sequences of the hand to fit or indeed physically improve upon kinematic data. Finally, we explore various visualization techniques

to present complex anatomical information of the hand in a form that is visually pleasing, and easily understandable, as inspired by plastination work of Dr. Gunther Von Hagen.

9.1 Future Work

In presenting the first skeletal musculo-tendon hand model with forward and inverse dynamics, we feel we have accomplished a small first step toward the realistic depiction of hand function. However, there is much that is still not understood about the human hand, leading to simplifications of our hand model. Thus, there is much that remains to be done for a construction of a more accurate anatomically based hand model.

9.1.1 A More Sophisticated Hand Model

In this work, there are simplifications made to the complex anatomy and biomechanics of the hand. Our model did not incorporate the nerve endings that activate a certain area of a muscle, thus neglecting the effects of having different areas on the muscle being stimulated. Instead, we assume that whole muscle contracts evenly and tendon is produced along its line of action, which are modeled as sequences of piecewise linear segments as it extends through one or more joints via tendon. This introduces inaccuracies since real insertion and origin attachment regions are not points, but areas. While this may be adequate when the real attachment region is small, as in the thread-like tendons through the wrist and fingers, the muscles' attachments 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 available on the distribution of muscle effort within 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.

For muscles that extend to tendons that spans over several digits, such as Flexor Digitorum Superficialis (FDS), Flexor Digitorum Profundus (FDP), and Extensor Digitorum (ED, the assumption that the whole muscle contracts evenly is inadequate. Thus, we introduced the notion of musculo-tendon unit, such that each of such muscle contains *four* musculo-tendon units corresponding to each tendon. The neurology of how these muscles contracts is ill-understood and differs among individuals. The contraction of related musculo-tendon units are interdependent, in which different areas of these muscles are more correlated and will stimulate simultaneously, resulting in the sympathetic joint motion. For example, the activation sites for the index finger is less correlated than between the other fingers, thus in general we can move our index finger more independently. Given our musculo-tendon unit model, we can model these muscles as each having 4 different activation sites. To produce sympathetic joint motion, we couple the four musculo-tendons units of the FDP and ED so that they are contracted simultaneously, while the four musculo-tendons of FPS units can be contracted independently. We are limited to such a simplistic approach until the interdependency can be somewhat quantified.

This interdependency introduces interesting areas of future work. Since interdependency varies amongst individuals, we could potentially motion capture different subjects performing a set of motion drills, and have the solver learn the parameters that best describes the interdependency for that particular individual. As a result, different hand models can be learnt for different subjects, including those suffering from hand problems, which will give insights on the reason why some people are more dexterous than others. There is also considerable room for optimism that our hand model can be run backward to develop families of hand models and morphologies for our ancestors, given fossil evidence.

In our model, external forces such as gravity are ignored for the sake of simplicity, allowing for real-time interactive performance, and for the computation the gradient of the objective

function. This can be added to get a more realistic representation of the physical environment. In addition, collision forces between muscles and bones can also be integrated to prevent interpenetration between digits.

The fundamental challenge of building any anatomically based model, whether for the hand or other body parts, is that it is impossible to create a generic model that fits all, since anatomical measurements vary greatly between individuals. Furthermore, there is much still unknown about the human body, especially the neurological aspect, to be able to create a complete anatomic model.

9.1.2 Validation and Calibration of Hand Model

The model requires validation against clinical data. To measure the activity of muscles, electromyography (EMG), a test that assesses the health of the muscles and the nerves controlling the muscles. A book by J.V. Basmajian [Bas79] contains a rich amount of information about EMG and hand muscles. For an EMG, a needle electrode is inserted through the skin into the muscle, and the electrical activity detected by this electrode is displayed on an oscilloscope, and may be heard through a speaker. Provided that such a test on human subjects is possible, we can record the muscle activities of subjects as we motion capture them performing motion drills. Then we can use the recorded data to validate the muscles activations from the inverse solution. Another approach in validation is to use the measured muscle activity levels as input to the forward simulation, and then measure the difference in the resulting motion from the motion captured.

Another improvement to our hand model is to measure the physical lengths of muscles tendons and moments, and bones of a real subject from MRI scans as done in [KM04]. This will generate more accurate results than using the average values obtained from cadavers, since individuals have varying muscles lengths and bone lengths.

The parameters of the muscles such as their strength, that are captured by Hill's three element model, are currently based on averaged values. Thus, these parameters needs to be calibrate against real data. One approach is to measure the error between the forward simulation motion when using generic muscle parameters, with the motion captured. Then we tweak the relative strength of the muscles, thereby, changing the amplitude of the blix curves, to see if are more accurate match is achieved.

9.1.3 Visualization

Better visualization techniques can be developed for better bone/muscle/joint interactions. For example, muscle and tendon can cause deformation to adjacent tissues such as other muscles and the overlying skin. As in [AHS03, Moc96, MTLT88], our hand can be augmented with a skin model for the deformation of the skin associated with the motion.

Appendix A

Hand Data

This appendix accumulates all relevant data from [AL99, BY94, BBT81, BH99] for the hand model.

A.1 Hand Joints and Bones Data

The limitations of the joints' rotations in our model are listed in Table A.1. The local coordinate systems of the joints have flexion/extension about the z-axis, and abduction/adduction about the y-axis. Opposition combines flexion and abduction, so that the thumb is folded over the palm, with its tips touching the pads of the fingers. Reposition is the reverse of opposition.

Table A.2 lists the length, radius, thickness and mass of the links described in Section 3. The values are shown in both unnormalized and normalized form. To normalize the length, we divided the measured length of the link by length of the hand, which is measured as the distance from the center of the radiocarpal joint to the tip of the third digit. The link's radius, width and thickness are normalized by dividing by the link's length. The mass of a link is assumed to be equal to the masses of corresponding bones plus all adjoining muscles, and divided by the

mass of the whole hand. These mass ratios are from [BY94].

Table A.1: Average Range of Motion for Hand Joints

Joint/DOF	Motion	Range (degree)
Distal Radiocarpal (Wrist)	Extension	0–70
	Flexion	0–80
	Radial Deviation	0–20
	Ulnar Deviation	0–30
Carpometacarpal (CMC) Thumb	Extension	0–20
	Flexion	0–15
	Abduction	0–70
	Adduction	0–20
	Opposition	Tip of thumb to pad of fifth digit
Metacarpophalangeal (MCP) Thumb	Flexion	0–50
Interphalangeal (IP) Thumb	Flexion	0–80
Metacarpophalangeal (MCP)	Extension	0–10
	Flexion	0–90
Index	Abduction	0–20
	Adduction	0–20
Middle	Abduction	0–10
	Adduction	0–10
Ring	Abduction	0–15
	Adduction	0–5
Pinky	Abduction	0–20
	Adduction	0–5
Proximal Interphalangeal (PIP)	Flexion	0–100
Distal Interphalangeal (DIP)	Extension	0–10
	Flexion	0–90

Table A.2: Lengths, Radius, Thickness, and Mass of Hand Bones

Joint/DOF	length (x)		radius (z)		thickness (y)		mass
	l (cm)	$\frac{l}{L}$	r (cm)	$\frac{r}{l}$	t (cm)	$\frac{t}{l}$	$\frac{m}{M}$
Wrist	19.77	0.451	6.575	0.333	3.445	0.174	0.662
CMC Thumb	11.396	0.26	1.766	0.155	1.856	0.163	0.121
MCP Thumb	7.588	0.173	1.52	0.200	1.837	0.242	0.22
IP Thumb	6.57	0.150	1.06	0.192	0.919	0.161	0.014
MCP Index	10.38	0.237	1.516	0.146	1.75	0.168	0.030
PIP Index	6.57	0.150	1.103	0.192	1.38	0.240	0.014
DIP Index	5.29	0.121	0.605	0.190	0.863	0.271	0.007
MCP Middle	11.12	0.254	1.534	0.136	2.10	0.190	0.033
PIP Middle	7.446	0.173	1.243	0.167	1.80	0.242	0.016
DIP Middle	5.29	0.121	0.724	0.190	1.12	0.294	0.007
MCP Ring	10.65	0.243	1.27	0.119	2.055	0.193	0.022
PIP Ring	7.35	0.168	1.165	0.172	1.750	0.258	0.015
DIP Ring	5.29	0.121	0.621	0.190	0.956	0.292	0.007
MCP Pinky	8.58	0.196	1.158	0.147	1.21	0.155	0.018
PIP Pinky	5.29	0.121	0.985	0.190	1.181	0.228	0.007
DIP Pinky	5.0	0.116	0.401	0.150	0.91	0.341	0.004

L=length of hand=43.83076923cm

M=mass of hand

A.2 Hand Muscles Data

Table A.3 shows the relative strength and the resting fiber lengths for principal hand muscles. These values are from Brand's study of relative muscle strength for the purpose of tendon transfer in [BBT81]

Table A.3: Hand Muscles Fiber Lengths and Tension Fractions

Muscle Name	Resting fiber length (cm)	Tension Fraction (%)
Flexor Carpi Radialis	5.2	4.1
Palmaris Longus	5.0	1.2
Flexor Digitorum Superficialis Index	7.2	2.0
Flexor Digitorum Superficialis Middle	7.0	3.4
Flexor Digitorum Superficialis Ring	7.3	2.0
Flexor Digitorum Superficialis Pinky	7.0	0.9
Flexor Carpi Ulnaris	6.7	4.2
Flexor Digitorum Profundus Index	6.6	2.7
Flexor Digitorum Profundus Middle	6.6	3.4
Flexor Digitorum Profundus Ring	6.8	3.0
Flexor Digitorum Profundus Pinky	6.2	2.8
Flexor Pollicis Longus	5.9	2.7
Extensor Carpi Radialis Longus	9.3	3.5
Extensor Carpi Radialis Brevis	6.1	4.2
Extensor Digitorum Index	5.5	1.0
Extensor Digitorum Middle	6.0	1.9
Extensor Digitorum Ring	5.8	1.7
Extensor Digitorum Pinky	5.9	0.9
Extensor Digiti Minimi	5.9	1.0
Extensor Carpi Ulnaris	4.5	4.5
Abductor Pollicis Longus	5.7	1.3
Extensor Indicis	5.5	1.0
Abductor Pollicis	4.6	3.1
Extensor Pollicis Brevis	4.3	0.8
Abductor Digiti Minimi	4.0	1.4
Flexor Digiti Minimi Brevis	3.4	0.4
Abductor Pollicis Brevis	3.7	1.1
Flexor Pollicis Brevis	3.6	1.3
Opponens Pollicis	2.4	1.9
Adductor Pollicis	3.6	3.0
Lumbrical I	5.5	0.2
Lumbrical II	6.6	0.2
Lumbrical III	6.0	0.1
Lumbrical IV	4.9	0.1
Palmar Interosseus	1.5	1.3
Palmar Interosseus II	1.7	1.2
Palmar Interosseus III	1.5	1.0
Dorsal Interosseus I	3.2	2.5
Dorsal Interosseus II	2.5	1.4
Dorsal Interosseus III	2.0	1.5
Dorsal Interosseus IV	1.7	1.5

Table A.4 lists the principal muscles of our system with the affected joints and the lever arms at the joint. The lever arm measurements can be found in [BH99]. Consider the configuration of the hand with the palm faced down, for the finger joints, the x-axis of the joint local co-ordinate is in the direction along the bone, the y-axis is pointing down, and the z-axis is pointing towards the 5th digit. Therefore, a positive lever arm in the y-axis produces extension, while a negative value produces flexion. Abduction/adduction is more complicated since abduction at the 2nd digit rotates in the same direction as adduction at the 4th and 5th digit. A positive lever arm in the z-axis produces adduction for the 2nd digit and abduction of the 4th and 5th digit. Likewise, a negative lever arm in the z-axis produces abduction for the 2nd digit and adduction of the 4th and 5th digit.

Table A.4: Hand Muscles Lever Arms

Muscle Name	Affecting Joint(s)/DOF	Lever Arm (cm)		
		x	y	z
FCR	wrist	0	1.75	-1.05
PL	wrist	0	2.1	-0.15
FDS II	wrist	0	1.5	0.3
	MCP Index	0	1.19	0.17
	PIP Index	0	0.62	0
FDS III	wrist	0	1.5	-0.15
	MCP Middle	0	1.19	0
	PIP Middle	0	0.62	0
FDS IV	wrist	0	1.5	0.52
	MCP Ring	0	1.19	-0.17
	PIP Ring	0	0.62	0
FDS V	wrist	0	1.5	0.75
	MCP Pinky	0	1.19	-0.17
	PIP Pinky	0	0.62	0
FCU	wrist	0	1.85	0.15
FDP II	wrist	0	0.6	-0.32
	MCP Index	0	1.11	0.6
	PIP Index	0	0.79	0
	DIP Index	0	0.41	0
FDP III	wrist	0	0.6	0.15
	MCP Middle	0	1.11	0
	PIP Middle	0	0.79	0
	DIP Middle	0	0.41	0
FDP IV	wrist	0	0.6	0.32
	MCP Ring	0	1.11	-0.6
	PIP Ring	0	0.79	0
	DIP Ring	0	0.41	0
FDP V	wrist	0	0.6	0.5
	MCP Pinky	0	1.11	-0.6
	PIP Pinky	0	0.79	0
	DIP Pinky	0	0.41	0

Muscle Name	Affecting Joint(s)/DOF	Lever Arm (cm)		
		x	y	z
FPL	wrist	0	1.3	-0.5
	CMC Thumb	0	1.0	1.0
	MCP Thumb	0	0.75	0
	IP Thumb	0	0.55	0
ECRL	wrist	0	-1.0	-2.1
ECRB	wrist	0	-1.3	-1.2
ED II	wrist	0	-1.3	-0.55
	MCP Index	0	-0.86	-0.02
	PIP Index	0	-0.28	0
	DIP Index	0	-0.22	0
ED III	wrist	0	-1.3	0
	MCP Middle	0	-0.86	-0.02
	PIP Middle	0	-0.28	0
	DIP Middle	0	-0.22	0
ED IV	wrist	0	-1.3	0.75
	MCP Ring	0	-0.86	-0.02
	PIP Ring	0	-0.28	0
	DIP Ring	0	-0.22	0
ED V	wrist	0	-1.3	1.0
	MCP Pinky	0	-0.86	-0.02
	PIP Pinky	0	-0.28	0
	DIP Pinky	0	-0.22	0
EDM	wrist	0	-1.3	0.75
	MCP Pinky	0	-0.86	0
	PIP Pinky	0	-0.26	0
	DIP Pinky	0	-0.19	0
ECU	wrist	0	-0.6	2.5
EPL	wrist	0	-0.9	-1.05
	CMC Thumb	0	-0.5	1.0
	MCP Thumb	0	-0.25	0
	IP Thumb	0	-0.2	0
EI	wrist	0	0.14	-0.04
	MCP Index	0	-0.9	0.13
	PIP Index	0	-0.26	0
	DIP Index	0	-0.19	0
APL	wrist	0	0.74	-2.4
	CMC Thumb	0	-0.5	0
EPB	wrist	0	-0.32	-2.3
	CMC Thumb	0	-0.45	-0.3
	MCP Thumb	0	-0.3	0
ADM	MCP Pinky	0	0	0.4
	PIP Pinky	0	-0.25	0
	DIP Pinky	0	0.2	0

Muscle Name	Affecting Joint(s)/DOF	Lever Arm (cm)		
		x	y	z
FDMB	MCP Pinky	0	0.4	0.4
APB	CMC Thumb	0	0.35	-0.75
	MCP Thumb	0	0.1	0
FPB	CMC Thumb	0	0.9	0.1
	MCP Thumb	0	0.7	0
OP	CMC Thumb	0	0.4	0.85
AP	CMC Thumb	0	0.45	0.9
	MCP Thumb	0	0.7	0
Lumb I	MCP Index	0	0.5	-0.48
	DIP Index	0	-0.18	0
	PIP Index	0	-0.07	0
Lumb II	MCP Middle	0	0.5	-0.48
	DIP Middle	0	-0.18	0
	PIP Middle	0	-0.07	0
Lumb III	MCP Ring	0	0.5	-0.48
	DIP Ring	0	-0.18	0
	PIP Ring	0	-0.07	0
Lumb IV	MCP Pinky	0	0.5	-0.48
	DIP Pinky	0	-0.18	0
	PIP Pinky	0	-0.07	0
PI I	MCP Index	0	0.66	0.58
	DIP Index	0	-0.26	0
	PIP Index	0	-0.16	0
PI II	MCP Ring	0	0.66	-0.58
	DIP Ring	0	-0.26	0
	PIP Ring	0	-0.16	0
PI III	MCP Pinky	0	0.66	-0.58
	DIP Pinky	0	-0.26	0
	PIP Pinky	0	-0.16	0
DI I	MCP Index	0	0.37	-0.61
	DIP Index	0	-0.26	0
	PIP Index	0	-0.16	0
DI II	MCP Middle	0	0.37	-0.61
	DIP Middle	0	-0.26	0
	PIP Middle	0	-0.16	0
DI III	MCP Middle	0	0.37	0.61
	DIP Middle	0	-0.26	0
	PIP Middle	0	-0.16	0
DI IV	MCP Pinky	0	0.37	0.61
	DIP Pinky	0	-0.26	0
	PIP Pinky	0	-0.16	0

Table A.5 shows the origins and insertions of the hand muscles. These points are measured from our 3D skeleton model with fitted digitalized muscle fibers and tendon data, by estimating the center of the area of attachments. These points are relative to our hand skeleton, and thus not to be taken as valid anatomical data. For each joint that the muscle or tendon passes through, we define origin and insertion points in the local co-ordinate system of its parent and itself respectively. The co-ordinates are normalized by dividing by the link's length, radius, and thickness (See Table A.2), for the x-axis, z-axis, and y-axis, respectively.

Table A.5: Muscles Origins and Insertions co-ordinates

Muscle Name	Affecting Joint(s)	Joint	Origin (cm)			Joint	Insertion (cm)		
			x	y	z		x	y	z
FCR	wrist	elbow	-0.04	0.11	0.34	wrist	0.37	0.47	-0.64
PL	wrist	elbow	0.04	0.76	2.07	wrist	0.51	0.60	0
FDS II	wrist	elbow	0.240	0.26	0.021	wrist	0	1.12	-0.15
	MCP II	wrist	0	1.12	-0.15	MCP II	0	1.02	0
	PIP II	MCP II	0	1.02	0	PIP II	0	0.58	0
FDS III	wrist	elbow	0.02	0.67	1.68	wrist	0	1.09	0.01
	MCP III	wrist	0	1.0	-0.25	MCP III	0	0.85	0
	PIP III	MCP III	0	0.85	0	PIP III	0	0.94	0
FDS IV	wrist	elbow	-0.08	0.77	2.35	wrist	0	0.9	0.21
	MCP IV	wrist	0	0.9	0.48	MCP IV	0	0.87	0
	PIP IV	MCP IV	0	0.87	0	PIP IV	0	0.73	0
FDS V	wrist	elbow	-0.08	0.77	2.35	wrist	0	0.88	0.38
	MCP V	wrist	0	0.88	0.71	MCP V	0	1.40	0
	PIP V	MCP V	0	1.40	0	PIP V	0	0.79	0
FCU	wrist	elbow	-0.04	0.78	2.39	wrist	0.24	0.77	0.59
FDP II	wrist	elbow	0.36	0.31	-0.28	wrist	0	0.91	-0.25
	MCP II	wrist	0	0.91	-0.54	MCP II	0	1.02	0
	PIP II	MCP II	0	1.02	0	PIP II	0	0.58	0
	DIP II	PIP II	0	0.58	0	DIP II	0	1.44	0
FDP III	wrist	elbow	0.41	0.31	0.75	wrist	0	1.03	0
	MCP III	wrist	0	1.09	-0.25	MCP III	0	0.85	0
	PIP III	MCP II	0	0.85	0	PIP III	0	0.94	0
	DIP III	PIP II	0	0.94	0	DIP III	0	1.04	0
FDP IV	wrist	elbow	0.41	0.31	0.75	wrist	0	0.84	0.22
	MCP IV	wrist	0	0.84	0.48	MCP IV	0	0.87	0
	PIP IV	MCP IV	0	0.87	0	PIP IV	0	0.73	0
	DIP IV	PIP IV	0	0.73	0	DIP IV	0	1.27	0

Muscle Name	Affecting Joint(s)	Joint	Origin (cm)			Joint	Insertion (cm)		
			x	y	z		x	y	z
FDP V	wrist	elbow	0.41	0.31	0.75	wrist	0	0.80	0.38
	MCP V	wrist	0	0.80	0.71	MCP V	0	1.40	0
	PIP V	MCP V	0	1.40	0	PIP V	0	0.79	0
	DIP V	PIP V	0	0.79	0	DIP V	0	1.27	0
FPL	wrist	elbow	0.62	0.09	0.23	wrist	0	1.03	-0.42
	CMC I	wrist	0.16	0.52	-0.73	CMC I	-0.03	1.37	1.25
	MCP I	CMC I	-0.03	1.37	1.25	MCP I	0	1.22	0
	IP I	MCP I	0	1.22	0	IP I	0	1.73	0
ECRL	wrist	elbow	-0.01	0.62	-0.81	wrist	0.39	-0.65	-0.70
ECRB	wrist	elbow	0.08	-0.25	-0.41	wrist	0.35	-0.65	-0.70
ED II	wrist	elbow	-0.01	-0.32	-0.22	wrist	0	-0.97	-0.35
	MCP II	wrist	0	-0.65	-0.246	MCP II	0	-0.6x	0
	PIP II	MCP II	0	-0.65	0	PIP II	0	-0.77	0
	DIP II	PIP II	0	-0.77	0	DIP II	0	-0.98	0
ED III	wrist	elbow	-0.01	0.32	-0.22	wrist	0	-1.05	-0.21
	MCP III	wrist	0	-1.05	-0.25	MCP III	0	-0.93	0
	PIP III	MCP III	0	-0.93	0	PIP III	0	-0.53	0
	DIP III	PIP III	0	-0.53	0	DIP III	0	-0.76	0
ED IV	wrist	elbow	-0.01	0.32	-0.22	wrist	0	-1.13	-0.07
	MCP IV	wrist	0	-1.13	0.38	MCP IV	0	-0.89	0
	PIP IV	MCP IV	0	-0.89	0	PIP IV	0	-0.61	0
	DIP IV	PIP IV	0	-0.61	0	DIP IV	0	-0.80	0
ED V	wrist	elbow	-0.01	0.32	-0.22	wrist	0	-1.17	-0.01
	MCP V	wrist	0	-1.17	0.79	MCP V	0	-1.56	0
	PIP V	MCP V	0	-1.56	0	PIP V	0	-0.86	0
	DIP V	PIP V	0	-0.87	0	DIP V	0	-0.82	0
EDM	wrist	elbow	-0.01	-0.32	-0.22	wrist	0	-1.15	-0.23
	MCP III	wrist	0	-0.65	0	MCP III	0	-1.56	0
	PIP III	MCP III	0	-1.56	0	PIP III	0	-0.86	0
	DIP III	PIP III	0	-0.86	0	DIP III	0	-0.82	0
ECU	wrist	elbow	0	-0.44	-0.15	wrist	0.08	-0.410	0.68
EPL	wrist	elbow	0.60	-0.42	0.12	wrist	0	-0.81	-0.4
	CMC I	wrist	0.16	-0.45	-0.7	CMC I	0	-0.64	1.17
	MCP I	CMC I	0	-0.64	1.17	MCP I	0	-0.97	0
	IP I	MCP I	0	-0.97	0	IP I	0	-1.48	0
EI	wrist	elbow	0.63	-0.13	0.29	wrist	0	-1.12	-0.24
	MCP II	wrist	0	-1.12	-0.24	MCP II	0	-0.65	0
	PIP II	MCP II	0	-0.65	0	PIP II	0	-0.77	0
	DIP II	PIP II	0	-0.77	0	DIP II	0	-0.98	0
APL	wrist	elbow	0.51	-0.35	-0.07	wrist	0	0.29	-1.24
	CMC I	wrist	0	0.29	-1.24	CMC I	0	-1.29	-0.51
EPB	wrist	elbow	0.79	-0.34	0.26	wrist	0	-0.35	-0.80
	CMC I	wrist	0	-0.35	-0.80	CMC I	0	-1.18	-0.12
	MCP I	CMC I	0	-1.18	-0.12	MCP I	0	-0.89	0
ADM	MCP V	wrist	0.28	0.49	0.56	MCP V	0	0	1.05
FDMB	MCP V	wrist	0.38	0.58	0.56	MCP V	0	0	1.05

Muscle Name	Affecting Joint(s)	Joint	Origin (cm)			Insertion (cm)			
			x	y	z	Joint	x	y	z
APB	CMC I	wrist	0.20	1.12	-0.45	CMC I	0	0.63	-0.41
	MCP I	CMC I	0	0.63	-0.41	MCP I	0	0.66	-1.24
FPB	CMC I	wrist	0.29	0.47	0.07	CMC I	0	1.5	0
	MCP I	CMC I	0	1.5	0	MCP I	0	0.67	-1.25
OP	CMC I	wrist	0.20	0.59	-0.22	CMC I	0	1.0	0
AP	CMC I	wrist	0.76	0.35	0	CMC I	0	1.99	1.94
		wrist	0.37	0.470	0				
	MCP I	CMC I	0.46	0.50	0.89	MCP I	0	0	1.10
Lumb I	MCP II	wrist	0.43	0.43	-0.55	MCP II	0	0.80	-1.13
	PIP II	MCP II	0.12	-0.84	0	PIP II	0	-0.65	0
	DIP II	PIP II	0	-0.65	0	DIP II	0	-0.71	0
Lumb II	MCP III	wrist	0.43	0.43	0	MCP III	0	0.80	-0.83
	PIP III	MCP III	0.14	-0.87	0	PIP III	0	-0.54	0
	DIP III	PIP III	0	-0.54	0	DIP III	0	-0.76	0
Lumb III	MCP IV	wrist	0.43	0.43	0	MCP IV	0	0.76	-1.08
		wrist	0.43	0.43	0.38				
	PIP IV	MCP IV	0.15	-1.54	0	PIP IV	0	-0.61	0
	DIP IV	PIP IV	0	-0.61	0	DIP IV	0	-0.80	0
Lumb IV	MCP V	wrist	0.43	0.43	0.38	MCP V	0	0.84	-1.21
		wrist	0.43	0.43	0.76				
	PIP V	MCP V	0.15	-1.07	0	PIP V	0	-0.87	0
	DIP V	PIP V	0	-0.87	0	DIP V	0	-0.82	0
PI I	MCP II	wrist	0.55	0.43	-0.40	MCP II	0	0	1.25
	PIP II	MCP II	0.12	-0.84	0	PIP II	0	-0.65	0
	DIP II	PIP II	0	-0.65	0	DIP II	0	-0.71	0
PI II	MCP IV	wrist	0.56	0.43	0.23	MCP IV	0	0	-1.40
	PIP IV	MCP II	0.15	-1.54	0	PIP IV	0	-0.61	0
	DIP IV	PIP II	0	-0.61	0	DIP IV	0	-0.8	0
PI III	MCP V	wrist	0.56	0.43	0.74	MCP V	0	0	1.40
	PIP V	MCP V	0.15	-1.07	0	PIP V	0	-0.87	0
	DIP V	PIP V	0	-0.87	0	DIP V	0	-0.82	0
DI I	MCP II	wrist	0.56	0	-0.79	MCP II	0.05	0	-1.45
		CMC I	0.07	0.80	0				
	PIP II	MCP II	0.12	-0.84	0	PIP II	0	-0.65	0
	DIP II	PIP II	0	-0.65	0	DIP II	0	-0.71	0
DI II	MCP III	wrist	0.56	0	-0.40	MCP II	0.09	0	-1.33
		wrist	0.56	0	-0.26				
	PIP III	MCP III	0.14	-0.87	0	PIP III	0	-0.53	0
	DIP III	PIP III	0	-0.53	0	DIP III	0	-0.76	0
DI III	MCP III	wrist	0.56	0	0.09	MCP II	0.09	0	-1.33
		wrist	0.56	0	0.22				
	PIP III	MCP III	0.14	-0.87	0	PIP III	0	-0.53	0
	DIP III	PIP III	0	-0.53	0	DIP III	0	-0.76	0
DI IV	MCP IV	wrist	0.56	0	0.56	MCP II	0.09	0	1.40
		wrist	0.56	0	0.74				
	PIP IV	MCP II	0.15	-1.54	0	PIP IV	0	-0.61	0
	DIP IV	PIP II	0	-0.61	0	DIP IV	0	-0.80	0

Appendix B

Inertia

B.1 The Inertia Tensor

Newton's first law of motion states "*A body maintains the current state of motion unless acted upon by an external force.*" This measure of the inertia in linear motion is the mass of the system and in angular motion is the moment of inertia. The moment of inertia of a body deals with how the mass is distributed throughout the body, thus bodies of the equal mass may possess different moments of inertia. The moment of inertia of a single particle is given by:

$$I = mr^2, \tag{B.1}$$

where m is the mass of the particle, and r is the perpendicular distance from the axis of rotation to the particle.

A rigid body can be thought of as a system of particles in which the relative positions of the particles do not change. Let \vec{r}_i be the displacement from the origin of the local coordinate system, then the inertia tensor rotating about an axis passing through the origin of the local reference frame is:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \sum_i \begin{bmatrix} m_i(r_{iy}^2 + r_{iz}^2) & -m_i r_{ix} r_{iy} & -m_i r_{ix} r_{iz} \\ -m_i r_{iy} r_{ix} & m_i(r_{ix}^2 + r_{iz}^2) & -m_i r_{iy} r_{iz} \\ -m_i r_{iz} r_{ix} & -m_i r_{iz} r_{iy} & m_i(r_{ix}^2 + r_{iy}^2) \end{bmatrix}.$$

The inertia tensor, \mathbf{I} , relates the angular momentum (\vec{L}) of a rigid body to its angular velocity ($\vec{\omega}$), analogous to how mass, \mathbf{M} , relates linear momentum (\vec{P}) and velocity (\vec{v}),

$$\mathbf{I} = \frac{\vec{L}}{\vec{\omega}} \quad \mathbf{M} = \frac{\vec{P}}{\vec{v}}.$$

The diagonal elements in the inertia tensor, I_{xx} , I_{yy} , and I_{zz} , are called the moments of inertia while the rest of the elements are called the products of inertia. Note that the inertia tensor is symmetrical.

B.2 The Transformation of Inertia Tensor

What happens when the rigid body's orientation changes during the simulation? Recomputing inertia tensor is expensive unless the body's shape is simple. By using the inertia tensor transformation properties and a pre-computed inertia tensor in the local coordinates space, we can compute the inertia tensor for an arbitrary rotation and translation.

B.2.1 Rotation

Let us first consider the effect of rotation. Suppose that the rotation matrix \mathbf{R} is applied to the rigid body. Let \vec{p}_i be the position of particle i in the local coordinates space, and \vec{c} be the position of the center of mass. Then the position of particle i in world coordinates becomes $\vec{x}_i = \mathbf{R}\vec{p}_i + \vec{c}$. The displacement from the center of mass is $\vec{r}_i = \mathbf{R}\vec{p}_i$. We can rewrite \mathbf{I} as:

$$\mathbf{I} = \sum_i m_i (r_{ix}^2 + r_{iy}^2 + r_{iz}^2) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} m_i r_{ix}^2 & m_i r_{ix} r_{iy} & -m_i r_{ix} r_{iz} \\ m_i r_{iy} r_{ix} & m_i r_{iy}^2 & m_i r_{iy} r_{iz} \\ m_i r_{iz} r_{ix} & m_i r_{iz} r_{iy} & m_i r_{iz}^2 \end{bmatrix}. \quad (\text{B.2})$$

Simplifying the above equation with the inner-product, $\vec{r}_i^T \vec{r}_i$ and outer-product, $\vec{r}_i \vec{r}_i^T$, and the identity matrix, \mathbf{e} , we get

$$\mathbf{I} = \sum_i m_i (\vec{r}_i^T \vec{r}_i \mathbf{e} - \vec{r}_i \vec{r}_i^T). \quad (\text{B.3})$$

Recall that the displacement from the center of mass is $\vec{r}_i = \mathbf{R} \vec{p}_i$, we further rewrite \mathbf{I} as:

$$\begin{aligned} \mathbf{I} &= \sum_i m_i [(\mathbf{R} \vec{p}_i)^T (\mathbf{R} \vec{p}_i) \mathbf{e} - (\mathbf{R} \vec{p}_i) (\mathbf{R} \vec{p}_i)^T] \\ &= \sum_i m_i [\vec{p}_i^T \vec{p}_i \mathbf{e} - \mathbf{R} \vec{p}_i \vec{p}_i^T \mathbf{R}^T] \\ &= \sum_i m_i [\mathbf{R} \vec{p}_i^T \vec{p}_i \mathbf{R}^T \mathbf{e} - \mathbf{R} \vec{p}_i \vec{p}_i^T \mathbf{R}^T] \\ &= \mathbf{R} \sum_i m_i [\vec{p}_i^T \vec{p}_i \mathbf{e} - \vec{p}_i \vec{p}_i^T] \mathbf{R}^T. \end{aligned} \quad (\text{B.4})$$

Defining \mathbf{I}_{body} to be inertia tensor in the local coordinate system of the body,

$$\mathbf{I}_{body} = \sum_i m_i (\vec{p}_i^T \vec{p}_i \mathbf{e} - \vec{p}_i \vec{p}_i^T). \quad (\text{B.5})$$

We can easily compute the inertia tensor, \mathbf{I} , rotated by \mathbf{R} , from \mathbf{I}_{body} by the *Similarity Transformation*:

$$\mathbf{I} = \mathbf{R} \mathbf{I}_{body} \mathbf{R}^T. \quad (\text{B.6})$$

B.2.2 Translation

Now, let's look at of translation. Consider two reference frames, $OXYZ$ system and the $O'X'Y'Z'$ system. The $OXYZ$ system is the local reference frame fixed to the body with its origin at the body's center of mass. The $O'X'Y'Z'$ system, is parallel to the $OXYZ$ system with

a different origin. We want to find the inertia tensor by translating the center of mass from the origin of $OXYZ$ system to the origin of $O'X'Y'Z'$ system. Let $\vec{a} = \vec{O} - \vec{O}'$ be that translation. The position of particle i in $OXYZ$ system is \vec{p}_i , thus its position in $O'X'Y'Z'$ system can be written as $\vec{r}_i = \vec{p}_i + \vec{a}$. The inertia tensor of the body about the $O'X'Y'Z'$ system is

$$\begin{aligned}
\mathbf{I} &= \sum_i \begin{bmatrix} m_i(r_{iy}^2 + r_{iz}^2) & -m_i r_{ix} r_{iy} & -m_i r_{ix} r_{iz} \\ -m_i r_{iy} r_{ix} & m_i(r_{ix}^2 + r_{iz}^2) & -m_i r_{iy} r_{iz} \\ -m_i r_{iz} r_{ix} & -m_i r_{iz} r_{iy} & m_i(r_{ix}^2 + r_{iy}^2) \end{bmatrix} \\
&= \sum_i m_i \begin{bmatrix} p_{iy}^2 + p_{iz}^2 & -p_{ix} p_{iy} & -p_{ix} p_{iz} \\ -p_{iy} p_{ix} & p_{ix}^2 + p_{iz}^2 & -p_{iy} p_{iz} \\ -p_{iz} p_{ix} & -p_{iz} p_{iy} & p_{ix}^2 + p_{iy}^2 \end{bmatrix} + M \begin{bmatrix} a_y^2 + a_z^2 & -a_x a_y & -a_x a_z \\ a_y a_x & a_x^2 + a_z^2 & -a_y a_z \\ -a_z a_x & -a_z a_y & a_x^2 + a_y^2 \end{bmatrix} \quad (\text{B.7}) \\
&= \sum_i m_i (\vec{p}_i^T \vec{p}_i \mathbf{e} - \vec{p}_i \vec{p}_i^T) + M (\vec{a}_i^T \vec{a}_i \mathbf{e} - \vec{a}_i \vec{a}_i^T) \\
&= \mathbf{I}_{body} + \mathbf{I}_t
\end{aligned}$$

where M is the mass of the body. \mathbf{I}_{body} is the inertia tensor of the body about its center of mass, and \mathbf{I}_t is the additional moment of inertia due to the translation of the reference frame.

B.3 The Inertia Tensor of Joints

The inertia tensor for all joints except the radiocarpal joint is approximated as one or a chain of cylinder(s). The bone at the end of the segment, i.e., one of the distal phalanges, is simply a cylinder of length l , radius r and mass m , rotated about an axis orthogonal to its length axis (x-axis) and passing through one of its ends. The inertia tensor of the body, \mathbf{I}_{body} , with the center of rotation at the origin of its local co-ordinate system is:

$$\mathbf{I}_{body} = \begin{bmatrix} \frac{1}{2}mr^2 & 0 & 0 \\ 0 & \frac{1}{4}mr^2 + \frac{1}{3}ml^2 & 0 \\ 0 & 0 & \frac{1}{4}mr^2 + \frac{1}{3}ml^2 \end{bmatrix}. \quad (\text{B.8})$$

Denote a hierarchy of cylinders as c_i, c_{i+1} corresponding to a hierarchy of joints as j_i, j_{i+1} , such that j_{i+1} is the child of j_i . When rotating about j_i , the total inertia tensor is composed of inertia tensors from c_i , and c_{i+1} in the local co-ordinates of the parent joint j_i . We denote the inertia tensor of the c_i in the local co-ordinates of j_i to be, \mathbf{I}_i^i , where the subscript indicates the co-ordinate system that the inertia tensor is in, and the superscript denotes the cylinder of interest. For cylinder c_i , the inertia tensor is simply a direct application of equation B.8,

$$\mathbf{I}_i^i = I_{body}^i. \quad (\text{B.9})$$

However, for cylinder c_{i+1} the axis of rotation does not pass through it, and thus its inertia tensor is not equal to I_{body} given in equation B.8. Instead, the inertia tensor is computed by using the rotation and translation properties described in the previous sections. We first apply a rotation, \mathbf{R} , so that the axis of rotation in the child's co-ordinate system aligns to the axis of rotation in the parent's co-ordinate system. Now that the co-ordinate systems are parallel, we can apply the translation property to translate the origin of the child's co-ordinate system to the origin of the parent's co-ordinate system.

Suppose we want to find the inertia tensor of c_{i+1} in the local co-ordinates of j_i . To find the change of basis matrix for the *Similarity Transformation*, we define the basis to be $\beta = \{\vec{r}_x, \vec{r}_y, \vec{r}_z\}$, where \vec{r}_n is the direction of the child's n-axis in the parent's co-ordinate system. Thus, the change of basis or the rotation matrix in Equation B.6 is

$$\mathbf{R} = \begin{bmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix}, \quad (\text{B.10})$$

where,

$$\vec{r}_x = \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}, \quad \vec{r}_y = \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}, \quad \vec{r}_z = \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix}. \quad (\text{B.11})$$

Using equation B.6 and equation B.7, we compute the inertia tensor of cylinder c_{i+1} in the co-ordinate system of parent to be,

$$\mathbf{I}_i^{i+1} = \mathbf{R}^{i+1} \mathbf{I}_{body}^{i+1} (\mathbf{R}^{i+1})^T + \mathbf{I}_i^{i+1}. \quad (\text{B.12})$$

The first term corresponds to the rotation of the frame of reference, and the second term corresponds to the translation of the frame of reference. The inertia tensor at j_i is found by summing up the inertia tensor from cylinder c_i and of the cylinders of its n children,

$$\mathbf{I}_i = \mathbf{I}_i^i + \sum_{j=1}^n \mathbf{I}_i^{i+j}. \quad (\text{B.13})$$

For the radiocarpal joint, a constant inertia tensor is assumed. This is pre-computed before simulation by using equation 7.1, by taking \vec{r}_i to be distances from the bone mesh vertices to the origin of the radiocarpal joint.

Appendix C

Gradient Calculation

In this section we illustrate the symbolic derivation of the gradient of the energy function, \vec{g} described in Chapter 6. The gradient is a m-dimensional vector for m parameters.

C.1 Gradients of Energy Functions

According to Chapter 6, our system has three energy functions from the state constraint, control constraint, number of muscle constraint. Using the *sum rule* we can separate the gradient into three terms, \vec{g}_s for the gradient of the motion evaluation term E_s , \vec{g}_c for the gradient of the controller evaluation term E_c , \vec{g}_m for the gradient of the number of muscle evaluation term E_m

$$\vec{g} = \vec{g}_s + \vec{g}_c + \vec{g}_m, \quad (\text{C.1})$$

\vec{g}_c is a m-dimensional vector that can be easily computed, since E_c is just the sum of the contractions values squared,

$$\vec{g}_c = w_c \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix}. \quad (\text{C.2})$$

g_m is also a m -dimensional vector that can be easily computed. The derivative of a unit step function is a unit impulse function (also known as the dirac delta function):

$$\frac{dU(c - c_o)}{dt} = \delta(c - c_o) = \begin{cases} \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon}, & c_o - \frac{\varepsilon}{2} \leq c \leq c_o + \frac{\varepsilon}{2} \\ 0, & \text{otherwise} \\ . & . \end{cases} \quad (\text{C.3})$$

Therefore, g_m is as follows, where c_o and ε are small positive number near zero:

$$\vec{g}_m = \frac{w_m}{m} \begin{pmatrix} \delta(c_1 - c_o) \\ \vdots \\ \delta(c_m - c_o) \end{pmatrix}. \quad (\text{C.4})$$

The computation of g_s is more involved. Using the *chain rule*, we can write the g_s as:

$$\begin{aligned} \vec{g}_s &= w_s \Delta_{\vec{c}} E(f(\vec{c})) \\ &= w_s \frac{\partial f(\vec{c})}{\partial \vec{c}}^T \cdot \Delta_{f(c)} E(f(\vec{c})) \end{aligned} \quad (\text{C.5})$$

The first term in Equation C.5 is a Jacobian matrix, containing the partial derivatives of the function with respect to its m parameters. Since $f: \mathfrak{R}^m \rightarrow \mathfrak{R}^{3n}$, the matrix has a dimension of $6n$ by m as shown below:

$$\begin{aligned} J(\vec{c}) &= \frac{\partial f(\vec{c})}{\partial \vec{c}} \\ &= \begin{pmatrix} \frac{\partial f}{\partial c_1} & \cdots & \frac{\partial f}{\partial c_m} \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial f_1}{\partial c_1} & \cdots & \frac{\partial f_1}{\partial c_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{6n}}{\partial c_1} & \cdots & \frac{\partial f_{6n}}{\partial c_m} \end{pmatrix} \end{aligned} \quad (\text{C.6})$$

To find the partial derivative of f , using equations of motion from Chapter 5, we express f_i for joint i as:

$$\begin{aligned} f_i(\vec{T}_i) &= \begin{pmatrix} r_t + \Delta t [\omega_t + \Delta t (I_i(t) + \Delta t k_\mu)^{-1} (\vec{T}_i(t) - k_\mu \omega_t)] \\ \omega_t + \Delta t (I_i(t) + \Delta t k_\mu)^{-1} (\vec{T}_i(t) - k_\mu \omega_t) \end{pmatrix} \\ \vec{T}_i(t) &= \sum_{j=1}^m \vec{r}_{i,j} \times (\alpha_j c_j F_{c,j}(l_j) + \alpha_j F_{s,j}(l_j)) \hat{F}_{i,j} \end{aligned} \quad (\text{C.7})$$

Since only the torque, $\vec{T}_i(t)$, is a function of \vec{c} , thus the partial derivative of f_i with respect to \vec{c} can be rewritten using the *chain rule*:

$$\frac{\partial f_i}{\partial \vec{c}} = \frac{\partial f_i}{\partial \vec{T}_i} \cdot \frac{\partial \vec{T}_i}{\partial \vec{c}}, \quad (\text{C.8})$$

where the partial derivative of the torque for joint i with respect to \vec{c} is a 3 by m matrix composed of cross products of the lever arm and the contraction force term:

$$\frac{\partial \vec{T}_i}{\partial \vec{c}} = \left(\alpha_1 F_{c,1}(l_1) [\vec{r}_{i,1} \times \hat{F}_{i,1}] \quad \dots \quad \alpha_m F_{c,m}(l_m) [\vec{r}_{i,m} \times \hat{F}_{i,m}] \right). \quad (\text{C.9})$$

The partial derivative of the f_i with respect to \vec{c} is a 6 by m matrix, corresponding to the $(6 * k)^{th}$ to $(6 * k + 5)^{th}$ rows for $k=0, \dots, n-1$ of the Jacobian matrix in Equation C.6,

$$\frac{\partial f_i}{\partial \vec{c}} = \begin{pmatrix} \Delta t^2 (I_i(t) + \Delta t k_\mu)^{-1} \frac{\partial T_i}{\partial \vec{c}} \\ \Delta t (I_i(t) + \Delta t k_\mu)^{-1} \frac{\partial T_i}{\partial \vec{c}} \end{pmatrix}. \quad (\text{C.10})$$

The next step after the computation of the Jacobian matrix is to find the second term in C.5. The partial derivative of the energy function with respect to the function $f(\vec{c})$ is a $6n$ by 1 vector,

$$\Delta_{f(c)} E(f(\vec{c})) = \begin{pmatrix} t_1 - y_1 \\ \vdots \\ t_{6n} - y_{6n} \end{pmatrix}. \quad (\text{C.11})$$

Finally, combining Equation C.2 and C.5, the gradient of the energy function is:

$$\vec{g} = w_c \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} + w_s J(\vec{c})^T \begin{pmatrix} t_1 - y_1 \\ \vdots \\ t_{6n} - y_{6n} \end{pmatrix}. \quad (\text{C.12})$$

This gradient, \vec{g} , along with the values of the energy functions, E_s , and E_c is compute multiple times per one time-step in the simulation of constrained system.

Bibliography

- [AHS03] I. Albrecht, J. Haber, and H. Seidel. Construction and animation of anatomically based human hand models. In *Proceedings of SIGGRAPH Symposium for Computer Animation 2003*, volume 22, pages 98–109. ACM Press / ACM SIGGRAPH, 2003.
- [AL99] A. M. R. Agur and M.J. Lee. *Grant's Atlas of Anatomy*. Lippincott Williams & Wilkins, Baltimore, Maryland, USA, 10 edition, 1999.
- [AP01a] F. C. Anderson and M. G. Pandy. Dynamic optimization of human walking. *Journal of Biomechanical Engineering*, 123(3):381–390, 2001.
- [AP01b] F. C. Anderson and M. G. Pandy. Static and dynamic optimization solutions for gait are practically equivalent. *Journal of Biomechanics*, 34:153–161, 2001.
- [Bar95] D. Baraff. An introduction to physically based modeling. ACM SIGGRAPH 1995 Course Notes, 1995.
- [Bar96] D. Baraff. Linear-time dynamics using lagrange multipliers. In *Proceedings of SIGGRAPH 1996*, volume 22 of *Computer Graphics Proceedings, Annual Conference Series*, pages 137–146. ACM Press / ACM SIGGRAPH, 1996.

- [Bar97] D. Baraff. Physically based modeling: Principles and practice *Implicit Methods for Differential Equations*. ACM SIGGRAPH 1997 Course Notes, 1997.
- [Bas79] J.V. Basmajian. *Muscles Alive. Their Functions Revealed by Electromyography*. Williams & Wilkins, Baltimore, Maryland, USA, 5 edition, 1979.
- [BB88] R. Barzel and A.J. Barr. A modeling system based on dynamic constraints. In *Proceedings of SIGGRAPH 1988*, volume 22, pages 309–315. ACM Press / ACM SIGGRAPH, 1988.
- [BBT81] P.W. Brand, R.B. Beach, and D.E. Thompson. Relative tension and potential excursion of muscles in the forearm and hand. *Journal of Hand Surgery*, 6(3):209–219, 1981.
- [BH99] P.W. Brand and A.M. Hollister. *Clinical Mechanics of the Hand*. Mosby - Year Book, Inc., St. Louis, MO, 3 edition, 1999.
- [BLN95] R.H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995.
- [BW92] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating flexible bodies. In *Proceedings of SIGGRAPH 1992*, volume 26, pages 303–311. ACM Press / ACM SIGGRAPH, 1992.
- [BY94] E. Biryukova and V. Yourovskaya. A model of hand dynamics. In *Advances in the Biomechanics of Hand and Wrist*, pages 107–122. Plenum Press, New York, 1994.
- [DL00] S. L. Delp and J. P. Loan. A computational framework for simulation and analysis of human and animal movement. volume 2, pages 46–55, 2000.

- [FP03] A. C. Fang and N. S. Pollard. Efficient synthesis of physically valid human motion. *ACM Trans. Graph.*, 22(3):417–426, 2003.
- [GE03] K. Singh G. ElKoura. Handrix: animating the human hand. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 110–119, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [GT95] R. Grzeszczuk and D. Terzopoulos. Automated learning of muscle-actuated locomotion through control abstraction. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 63–70, New York, NY, USA, 1995. ACM Press.
- [GTH98] R. Grzeszczuk, D. Terzopoulos, and G. Hinton. Neuroanimator: fast neural network emulation and control of physics-based models. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 9–20, New York, NY, USA, 1998. ACM Press.
- [Hag] G. V. Hagen. Body worlds, the anatomical exhibition of real human bodies.
- [HBR⁺92] K.H. Hohne, M. Bomans, M. Riemer, R. Schubert, U. Tiede, and W. Lierse. A volume-based anatomical atlas. *IEEE Comput. Graph. Appl.*, 12(4):73–78, 1992.
- [Hil38] A.V. Hill. The heat of shortening and dynamic constraints of muscle. volume 126, pages 136–195, 1938.
- [HZG90] M.G. Hoy, F.E. Zajac, and M.E. Gordon. A musculoskeletal model of the human lower extremity: the effect of muscle, tendon, and moment arm on the moment-angle relationship of musculotendon actuators at the hip, knee, and ankle. *J. Biomechanics*, 23(2):157–169, 1990.

- [IC87] P. Isaacs and M. Cohen. Controlling dynamic simulation with kinematic constraints. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 215–224, New York, NY, USA, 1987. ACM Press.
- [KGP02] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 473–482, New York, NY, USA, 2002. ACM Press.
- [KJP02] P.G. Kry, D.L. James, and D.K. Pai. EigenSkin: Real time large deformation character skinning in hardware. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 153–159, July 2002.
- [KM04] T. Kurihara and N. Miyata. Modeling deformable human hands from medical images. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 355–363. ACM Press, 2004.
- [LK95] J. Lee and T. Kunii. Model-based analysis of hand posture. In *IEEE Computer Graphics and Applications*, volume 15, pages 77–86, 1995.
- [Mil88] G.S.P. Miller. The motion dynamics of snakes and worms. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 169–173, New York, NY, USA, 1988. ACM Press.
- [MM00] M.W. Marzke and R.F. Marzke. Evolution of the human hand: approaches to acquiring, analysing and interpreting the anatomical evidence. *Journal of Anatomy*, 197:121–140, 2000.
- [Moc96] L. Moccozet. *Hand Modeling and Animation for Virtual Humans*. PhD thesis, University of Geneva, 1996.

- [MTA⁺01] J. McDonald, J. Toro, K. Alkoby, A. Berthiaume, R. Carter, P. Chomwong, J. Christopher, M. Davidson, R. Furst, B. Konie, G. Lancaster, L. Roychoudhuri, E. Sedgewick, N. Tomuro, and R. Wolfe. An improved articulated model of the human hand. *The Visual Computer*, 17(3):158–166, 2001.
- [MTB03] M.J. McGuffin, L. Tancou, and R. Balakrishnan. Using deformation for browsing volumetric data. In *Proceedings of IEEE Visualization*, pages 401–408. IEEE Computer Society Press, 2003.
- [MTLT88] N. Magnenat-Thalmann, R. Laperriere, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface 1988*, pages 29–33. ACM Press / ACM SIGGRAPH, 1988.
- [NTH01] V. Ng-Thow-Hing. *Anatomically-Based Models for Physical and Geometric Reconstruction of Humans and Other Animals*. PhD thesis, University of Toronto, 2001.
- [NTHF02] V. Ng-Thow-Hing and E. Fiume. Application-Specific Muscle Representations. In *Proc. Graphics Interface*, pages 107–116, May 2002.
- [Pai02] D.K. Pai. Strands: Interactive simulation of thin solids using cosserat models. *Proceedings of Eurographics 2002, Computer Graphics Forum*, 21(3):347–352, September 2002.
- [PB88] J. C. Platt and A. H. Barr. Constraints methods for flexible models. pages 279–288, 1988.
- [PB02] K. Pullen and C. Bregler. Motion capture assisted animation: texturing and synthesis. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 501–508, New York, NY, USA, 2002. ACM Press.

- [PTVF02] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C++, 2nd Edition*. The Press Syndicate of the University of Cambridge, Cambridge, UK, 2002.
- [RG91] H. Rijkema and M. Girard. Computer animation of knowledge-based human grasping. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 339–348, New York, NY, USA, 1991. ACM Press.
- [TAD03] D. G. Thelne, F. C. Anderson, and S. L. Delp. Generating dynamic simulations of movement using computed muscle control. volume 36, pages 321–328, 2003.
- [TT94] X. Tu and D. Terzopoulos. Artificial fishes: physics, locomotion, perception, behavior. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 43–50, New York, NY, USA, 1994. ACM Press.
- [WFB87] A. Witkin, K. Fleischer, and A. Barr. Energy constraints on parameterized models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 225–232, New York, NY, USA, 1987. ACM Press.
- [Wit01] A. Witkin. Physically based modelling: Constrained dynamics. ACM SIGGRAPH 2001 Course Notes, 2001.
- [WK88] A. Witkin and M. Kass. Spacetime constraints. In *In Proceedings of the 1988 ACM SIGGRAPH*, volume 22, pages 149–167. ACM Press / ACM SIGGRAPH, 1988.
- [Zaj89] F. E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*,

17:359–411, 1989.

- [ZBLN97] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, 1997.