Pergamon

Comput. Biol. Med. Vol. 25, No. 1, pp. 21–34, 1995
Copyright © 1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0010–4825/95 $9.50 + 0.00

0010-4825(94)00053-0

A GRAPHICS-BASED SOFTWARE SYSTEM TO DEVELOP AND ANALYZE MODELS OF MUSCULOSKELETAL STRUCTURES

SCOTT L. DELP* and J. PETER LOAN†

* Departments of Biomedical Engineering and Physical Medicine & Rehabilitation, Northwestern University, and Sensory Motor Performance Program, Rehabilitation Institute of Chicago, Chicago, IL 60611, U.S.A.; and, † MusculoGraphics Inc, Northwestern University Research Park, Evanston, IL 60201, U.S.A.

(*Received* 6 *April* 1994; *received in revised form* 27 *October* 1994)

Abstract—We have created a graphics-based software system that enables users to develop and analyze musculoskeletal models without programming. To define a model using this system one specifies the surfaces of the bones, the kinematics of the joints and the lines of action and force-generating parameters of the muscles. Once a model is defined, the function of each muscle can be analyzed by computing its length, moment arms, force and joint moments. The software has been implemented on a computer graphics workstation so that users can view the model from any perspective and graphically manipulate the joint kinematics and musculoskeletal geometry. Models can also be animated to visualize the results of motion analysis experiments. Since the software can be used to study models of many different musculoskeletal structures, it can enhance the productivity of investigators working on diverse problems in biomechanics.

Biomechanics Computer modeling Computer graphics
Surgical simulation Animation Musculoskeletal Muscle

INTRODUCTION

With the recent advances in medical imaging technology, numerous graphics-based software packages have been developed to process, display and manipulate medical images. For example, Robb *et al.* have created a software system to manipulate images from computed tomography (CT), magnetic resonance (MR), and other imaging modalities [1]. Such computer-based systems are now being used in a wide variety of medical applications [2], including analysis of the musculoskeletal system [3]. For instance, three-dimensional (3D) reconstructions from CT scans have been used to design orthopaedic implants [4] and plan orthopaedic surgeries [5]. The ability to visualize the complex geometric relationships among the muscles, bones, and other structures is the main advantage of using 3D graphics to analyze the musculoskeletal system. Analysis of 3D images alone, however, provides limited insight into the function of the muscles and joints because the images do not quantify muscle moment arms, ligament strains, joint motions, or other factors that affect musculoskeletal function.

Biomechanics researchers are interested in muscle forces and how the geometric relationships among the muscles and bones transform these forces into moments about the joints. As a result, mathematical models of many different musculoskeletal structures have been developed to examine these issues. For example, models of the human hip [6, 7] and lower limb [8] have been developed to study the effects of musculoskeletal surgeries on the moment-generating capacities of the muscles. Models of the upper extremity have been developed to quantify musculoskeletal anatomy [9], design tendon transfer surgeries [10], and investigate motor control in static arm postures [11]. Mai and Lieber developed a model of the frog hindlimb to investigate the force–length characteristics of the semitendinosus muscle [12]. Wickland *et al.* analyzed origins and insertions of cat neck muscles to quantify their moment-generating properties [13].

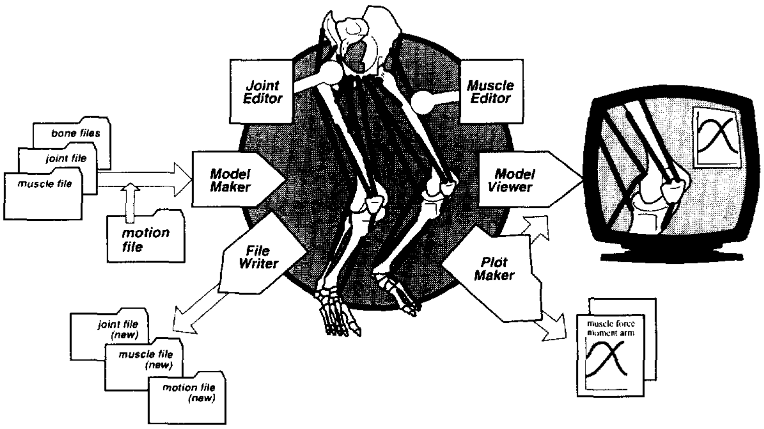


Fig. 1. Structure of the musculoskeletal modeling software. Input files describing the bone surfaces (bone files), joint kinematics (joint file), and muscle–tendon parameters (muscle file) are read in to make a musculoskeletal model. A model can be altered using the joint editor, muscle editor and model viewer. Information is extracted from the model by making plots or by exporting edited joint and muscle files. A motion file containing the joint angles that describe a movement can be loaded to create an animation of the movement.

Although numerous musculoskeletal models have been developed, a standard for representing these models in computer software has not been described. Thus, it is difficult for a musculoskeletal model to be evaluated or used outside the laboratory where it is developed. Also, since software tools are not generally available to assist in the development of musculoskeletal models, researchers typically must spend a great deal of time implementing each new model in software.

We have created a software package called SIMM (Software for Interactive Musculoskeletal Modeling) that enables users to develop, alter, and evaluate models of many different musculoskeletal structures, without programming. The purpose of SIMM is to help users create models that quantify the effects of musculoskeletal geometry, joint kinematics, and muscle-tendon parameters on muscle-tendon lengths, moment arms, muscle forces and joint moments. Since the software is implemented on a computer graphics workstation, users can visualize the musculoskeletal geometry and interact with the models.

This article describes the musculoskeletal modeling software, demonstrates how to define and interact with musculoskeletal models, discusses several applications of the software, and points out some of its limitations.

SOFTWARE OVERVIEW

The software was developed using the C language, runs on Silicon Graphics (Mountain View, CA) workstations, and requires at least 16MB of memory. Four goals were established when designing and implementing SIMM. Specifically, the software should:

(i) be *general* enough so that a wide variety of musculoskeletal structures can be modeled,
(ii) provide *realistic* models of muscle, tendon, and ligaments, and allow accurate specification of joint kinematics,
(iii) provide an *interactive* graphics-based environment so the model can be visualized, altered, tested, and analyzed efficiently, and
(iv) be *extensible* so that new capabilities can be added to the software.

SIMM allows the user to load one or more musculoskeletal models by reading a set of bone files, joint files, and muscle files (Fig. 1). The "model maker" scans these input files and creates a data structure that represents the musculoskeletal model. Once loaded, a model can be acted upon by a number of editing and analysis tools. Each tool has a distinct function, such as altering the muscles or joints, or making plots for analysis.

DEFINING A MUSCULOSKELETAL MODEL

In SIMM, a model consists of a set of rigid body segments that are connected by joints. Muscle-tendon actuators span the joints and develop force, thus generating moments about the joints. SIMM enables an analysis of a musculoskeletal model by calculating the length and moment arm of each modeled actuator. Also, given muscle activation, the force and joint moments that each actuator generates can be computed over a range of body positions.

Body segments

A body segment consists of a reference frame that contains one or more bones. Each bone is stored in a file that lists the polygons comprising the bone surface. The vertices of the polygons are expressed in the Cartesian coordinate system that serves as the reference frame for the body segment. These reference frames are used to specify the attachments of muscles and ligaments and to define the position and orientation of one body segment with respect to another.

System kinematics

Body segments can be connected in any arrangement by defining joints. A joint specifies the transformations that relate the position and orientation of one body segment to another. The transformations are performed in a user-defined order, and consist of three orthogonal translations and three rotations around user-defined axes. Each translation and rotation is either constant (i.e. independent of body position) or a function of a generalized coordinate. A generalized coordinate defines a degree of freedom in the model (e.g. a joint angle). "Kinematic functions" relate the translations and rotations to the generalized coordinates.

The joint file defines the joints and the kinematic functions used in the joint definitions. For example, in a model of the upper extremity, the elbow joint definition specifies the transformations from the HUMERUS segment to the ULNA segment (see below). The order of the transformations indicates that to move from the HUMERUS reference frame to the ULNA reference frame, a translation is performed first, followed by rotations around axis1, axis2, and axis3, respectively. The translation is broken into three orthogonal components (tx, ty, tz), where the *x*, *y* and *z* directions are defined by the axes of the HUMERUS reference frame. The rotation axes are also defined relative to the HUMERUS reference frame, and their orientations can be changed by adjusting the axis definitions. Since all of the translations and rotations in this joint are either constants or functions of a single generalized coordinate (elbow_angle), the joint has only one degree of freedom. It is not a revolute joint, however, because two of the translations (tx, ty) are functions of elbow_angle.

```
/* ELBOW JOINT DEFINITION */    /*                  comments                       */
beginjoint elbow                /* joint from HUMERUS frame to ULNA frame          */
segments HUMERUS ULNA
order t r1 r2 r3                /* translate, then rotate about axis1,
                                                 axis2, and axis3                  */
axis1 1.0   0.0   0.0           /* rotation axis definitions: axes are defined     */
axis2 0.0   1.0   0.0           /* with respect to the humerus coordinate system   */
axis3 0.22  0.19  0.95
tx function f1 (elbow_angle)    /* the kinematic functions f1 and f2 specify       */
ty function f2 (elbow_angle)    /* the relations between the generalized           */
tz constant 0.04                /* coordinate (elbow_angle) and tx and ty,         */
r1 constant 0.00                /* the translations in the anterior-posterior      */
r2 constant 0.00                /* and superior-inferior directions, respectively. */
r3 function f3 (elbow_angle)    /* f3 specifies the relation between elbow_angle   */
endjoint                        /* and the rotation about axis3, the elbow         */
                                /* rotation axis.                                  */
```

Kinematic functions define translations and rotations between body segments as functions of the generalized coordinates. These functions can be derived from biomechanical investigations of joint motion. For example, Chao and Morrey report the location of the elbow axis of rotation, which determines tx and ty, as a function of elbow angle [14]. The relationship between tx and elbow_angle is specified by the kinematic function, f1, shown below. The (elbow_angle, tx) pairs are interpolated by a natural cubic spline using an algorithm by Forsythe and Malcolm [15]. Thus, the value of tx can be determined by evaluating the cubic spline at any elbow_angle.

```
/* EXAMPLE KINEMATIC FUNCTION */
beginfunction f1              /* defines the x translation from humerus to ulna */
(0.0,  0.015)                 /* elbow_angle (degrees), tx (meters) */
(8.0,  0.013)
(17.0, 0.012)
(35.0, 0.011)
(66.0, 0.011)
(130, 0.011)
endfunction
```

Muscle-tendon model

A muscle-tendon actuator is defined by specifying its geometry and force-generating properties. The geometry of a muscle-tendon actuator is defined by a series of points, which are connected by line segments. Each point is fixed to one of the body segments, and is expressed in that segment's reference frame. A minimum of two points is required to define the muscle path. However, any number of "via" points can be used to describe a muscle path that wraps over bone or is constrained by retinacula. By specifying a restricted range of joint angles over which these via points constrain the path, the number of line segments in the muscle path can change with joint position.

The force-generating properties of a specific muscle-tendon actuator are derived by scaling a generic, Hill-type model [16]. To create a muscle model, the user must supply five parameters and four curves (Fig. 2). The five parameters are: peak isometric muscle force (F_o^M), optimal muscle-fiber length (ℓ_o^M), pennation angle at optimal fiber length

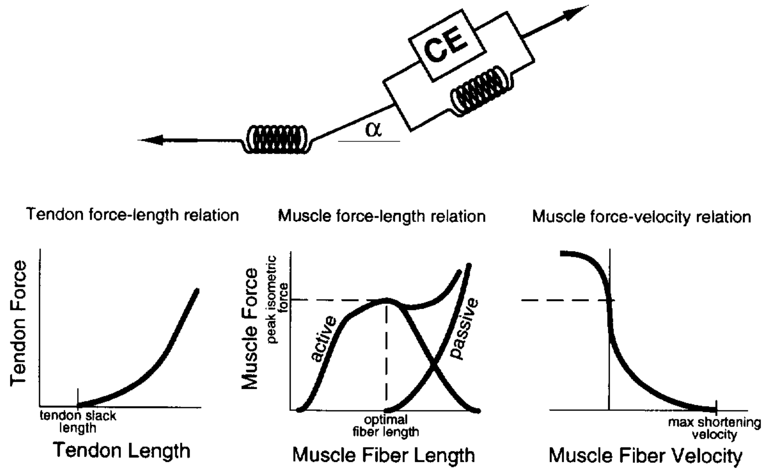


Fig. 2. Muscle-tendon actuator model. The properties of muscle are represented by an active contractile element (CE) in parallel with a passive elastic element. Muscle is in series with tendon, which is represented by a non-linear elastic element. α, the pennation angle, is the angle between the muscle fibers and the tendon. Muscle force is assumed to be the sum of active and passive forces. The forces in muscle and tendon are normalized by peak isometric muscle force. Muscle-fiber length is normalized by optimal muscle-fiber length. For a given muscle-tendon length, velocity, and activation level, the model computes muscle and tendon forces (see text for details).

(α_o), tendon slack length (ℓ_s^T), and maximum shortening velocity (V_m). The four curves are the force–length relation of tendon, active and passive force–length relations of muscle, and force–velocity relation of muscle. The active force–length curve of muscle is scaled linearly by muscle activation, which ranges from 0 (no activation) to 1 (full activation) [16]. All of the curves are defined by listing control points which are interpolated by natural cubic splines [15].

The muscle file contains definitions of each muscle-tendon actuator. A "default" muscle can be defined which contains the parameters that are usually identical for every muscle (e.g. normalized force–length curves). If a particular muscle definition does not contain a certain parameter or curve, the parameter or curve is inherited from the default muscle. This greatly reduces the number of parameters that must be specified in each muscle definition.

A sample muscle, BicepsBrachii, is defined below. The muscle path is defined by four points, one in the SCAPULA reference frame, two in the HUMERUS reference frame, and one in the RADIUS reference frame. The second point in the HUMERUS frame is included in the muscle path only when elbow_angle is between 0 and 30 degrees. This is done so that the muscle wraps over the distal humerus when the elbow is extended beyond 30 degrees. The muscle is a member of three groups, shoulder_flexors, elbow_flexors and supinators, which means that the muscle name will appear in those three muscle-group menus on the workstation display. The muscle definition also lists the values of the five muscle-tendon parameters (F_o^M, ℓ_o^M, ℓ_s^T, α_o, V_m). Since the curves that specify the force–length–velocity relations of muscle and tendon are not given in this example, they would be inherited from the default muscle.

```
/* EXAMPLE MUSCLE */
beginmuscle BicepsBrachii
beginpoints
   0.013  -0.038    -0.029    segment  SCAPULA
   0.029  -0.239     0.031    segment  HUMERUS
   0.031  -0.267     0.008    segment  HUMERUS   range elbow_angle (0.0, 30.0)
  -0.002  -0.030     0.003    segment  RADIUS
endpoints
begininggroups          shoulder_flexors          elbow_flexors  supinators  endgroups
force_max                          320.0              /* Newtons */
optimal_fiber_length                 0.12             /* meters */
tendon_slack_length                  0.17             /* meters */
pennation_angle                      0.0              /* degrees */
max_fiber_velocity                   7.0              /* fiber lengths/sec */
endmuscle
```

COMPUTING MOMENT ARM, LENGTH AND FORCE

For every joint that a muscle spans, the muscle has a **moment arm** for each generalized coordinate in the joint definition. For example, because the BicepsBrachii muscle, defined above, spans the elbow, shoulder, and radial-ulnar joint, it has a moment arm for elbow_angle and for each generalized coordinate (degree of freedom) in the shoulder and the radial-ulnar joints. We calculate moment arms using the "partial velocity" method [38]. In this method, the partial velocity of point P_m in reference frame A with respect to a generalized coordinate q_i is defined as the partial derivative of the position of P_m in A with respect to q_i (i.e. $\partial p/\partial q_i$, see Fig. 3). Equations (1)–(4) determine the moment arm of a muscle for a generalized coordinate q_i as follows (vectors are bold).

First, let $\boldsymbol{u}$, the translational component of the partial velocity, be defined as:

$$\boldsymbol{u} = (\partial tx/\partial q_i)\boldsymbol{x} + (\partial ty/\partial q_i)\boldsymbol{y} + (\partial tz/\partial q_i)\boldsymbol{z} \qquad (1)$$

where tx, ty and tz are the constants or kinematic functions that define the translations

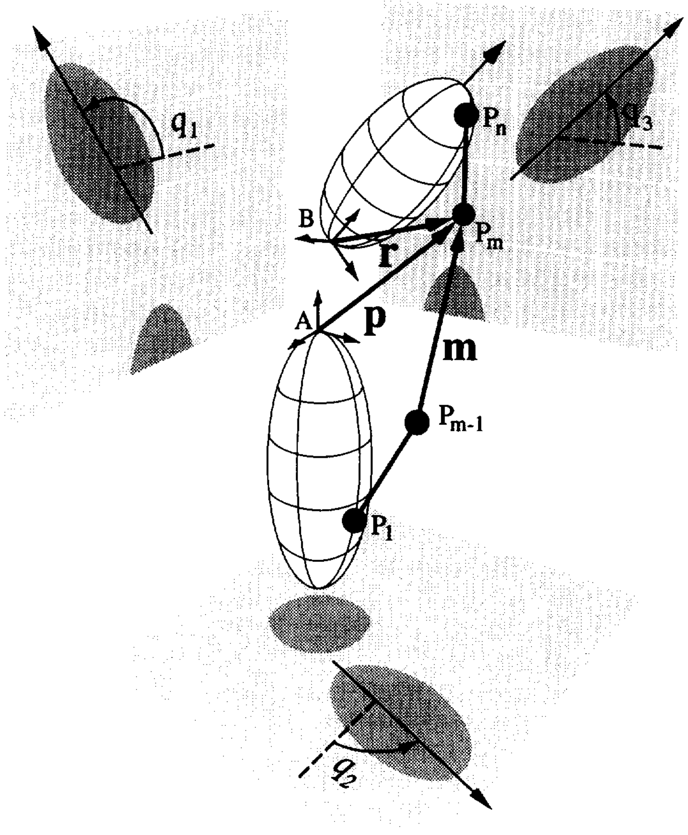


Fig. 3. Definition of terms used in moment arm calculations. Points P_1–P_n define the muscle path. P_1–P_{m-1} are fixed in body A. P_m–P_n are fixed in body B. $\boldsymbol{m}$ is the vector from point P_{m-1} to P_m. $\boldsymbol{P}$ expresses point P_m in reference frame A. r expresses P_m in B. In general, six generalized coordinates (three rotations angles: q_1, q_2, q_3, and three translation coordinates, not shown) are needed to characterize the orientation and position of body B relative to body A. The moment arm for each generalized coordinate (q_i) is given by $\partial p/\partial q_i \cdot \boldsymbol{m}$.

between the two body segments. Next, let ω, the angular component of the partial velocity, be defined as:

$$\boldsymbol{\omega} = (\partial r1/\partial q_i)\boldsymbol{\lambda}_1 + (\partial r2/\partial q_i)\boldsymbol{\lambda}_2 + (\partial r3/\partial q_i)\boldsymbol{\lambda}_3, \qquad (2)$$

where $r1$, $r2$ and $r3$ are the constants or kinematic functions that define the rotations about $\boldsymbol{\lambda}_1$ (axis1), $\boldsymbol{\lambda}_2$ (axis2) and $\boldsymbol{\lambda}_3$ (axis3), respectively. Then, ${}^{A}V^{P_m}$, the partial velocity of the muscle point P_m in reference frame A, is:

$$ {}^{A}V^{P_m} = \boldsymbol{u} + (\boldsymbol{\omega} \times \boldsymbol{r}) \qquad (3)$$

where $\boldsymbol{r}$ is P_m in reference frame B (Fig. 3). Finally, the moment arm (ma_i) of the muscle for the generalized coordinate q_i is given by:

$$ma_i = {}^{A}V^{P_m} \cdot \boldsymbol{m}, \qquad (4)$$

where $\boldsymbol{m}$ is a normalized vector along the muscle line of action.

The advantage of the partial velocity method is that it provides a consistent technique to compute the moment arms of muscles crossing all types of joints. Thus, the same algorithm can be used to calculate the moment arms of muscles about the knee, hip, elbow and shoulder, even though the mechanics of these joints differ considerably. The partial velocity method is equivalent to calculating a moment arm (ma_i) as the change in muscle-tendon length $(\partial \ell^{\mathrm{MT}})$ with respect to a change in the generalized coordinate (∂q_i). That is, (4) is equivalent to

$$ma_i = (\partial \ell^{\mathrm{MT}}/\partial q_i). \qquad (5)$$

Although equation (5) provides an excellent method to compute moment arms from experimental measurements of the change in ℓ^{MT} with joint angle [17], equation (4) is more convenient to use in musculoskeletal modeling, because moment arms can be determined analytically, given only muscle attachment coordinates ($P_1 \ldots P_n$) and joint kinematics.

Muscle-tendon **length** is calculated as the sum of the lengths of the line segments that define the muscle path. That is, if a muscle path is defined by points P_1–P_n, then muscle-tendon length (ℓ^{MT}) is determined by transforming the muscle points to a common reference frame and then computing

$$\ell^{\mathrm{MT}} = \sum_{1 \leq i < n} |P_{i+1} - P_i|. \tag{6}$$

Muscle-tendon **velocity** (V^{MT}) is computed as the sum of the velocities of the $n-1$ line segments that define the muscle path. That is,

$$V^{\mathrm{MT}} = \sum_{1 \leq j < n} V_j. \tag{7}$$

V_j, the velocity of an individual line segment, is computed as

$$V_j = \sum_{1 \leq i \leq r} ma_i \left(\frac{\mathrm{d}q_i}{\mathrm{d}t} \right), \tag{8}$$

where r is the number of generalized coordinates for which the muscle has a moment arm, ma_i is computed using equation (4), and $\mathrm{d}q_i/\mathrm{d}t$ is the time derivative of the generalized coordinate q_i.

Once ℓ^{MT} and V^{MT} are determined, muscle **force** can be estimated with the following iterative algorithm. An initial value of the muscle-fiber length and tendon length is assumed based on the muscle's optimal fiber length and ℓ^{MT}. The force generated by active muscle is calculated using the assumed fiber length to interpolate the active force–length curve, which is scaled by activation and then modified by force–velocity effects. The fiber velocity is a fraction of V^{MT}; the fraction is determined by the relative stiffnesses of the muscle and its tendon. The passive force is then estimated based on the muscle fiber length and added to the active force. The force in the tendon is found by using its length to interpolate the force–length curve of the tendon. The force in tendon (F^{T}) and muscle (F^{M}) are related by

$$F^{\mathrm{T}} = F^{\mathrm{M}}(\cos \alpha), \tag{9}$$

where α is the pennation angle [16]. If the computed muscle and tendon forces satisfy equation (9), then the muscle and tendon forces have been found. Otherwise, the muscle and tendon lengths are adjusted based on the slopes of their force–length curves at their respective lengths, and the process is repeated. This algorithm generally converges on a solution within four iterations.

MOTION FILES AND ANIMATION

The user may create motion files, which contain a sequence of joint angles describing a movement. Once loaded into the program, motion files can be used to simultaneously vary multiple degrees of freedom, animate the model display, and provide new independent variables for generating plots. For example, a model of the human lower limbs can be animated using data collected in a gait analysis laboratory (Fig. 4). This allows one to plot muscle-tendon lengths and moment arms vs percent of gait cycle. If the motion file contains activation levels for each muscle, say from normalized, low-pass filtered electromyographic (EMG) data, the animation displays these activation levels by varying muscle color and thickness. We have found that the animation capability, including the display of processed EMG data, is an effective tool for interpreting the results of motion studies.

THE USER INTERFACE

The user interface consists of graphical tools that help to specify joint kinematics and muscle attachments, compare model parameters with experimental data, and execute sensitivity studies.

The model viewer allows the user to view models from any perspective and manipulate the body into various configurations. Individual joints can be manipulated using slider bars or by entering values of the generalized coordinates with the keyboard. Muscles can be turned on and off to examine their relative locations.

The joint editor allows the user to graphically manipulate the kinematics of joints. Once a joint is selected, the three rotations (*r*1, *r*2, *r*3) and three translations (*tx*, *ty*, *tz*) that comprise the joint are displayed. If the rotations and translations are constants, they can be changed by typing in a new value. If they are functions of the generalized coordinates, they can be changed by moving, adding, or deleting control points of the cubic splines that define the kinematic functions. For example, when developing a model of the human upper limb, the user may choose to alter the kinematic functions that define the relative motion of the humerus and ulna (Fig. 5). This may be done to refine kinematic functions or to study the sensitivity of elbow flexion moment arms to joint modeling assumptions. For instance, one may compare moment arms calculated using a fixed joint center of rotation to moment arms with a moving joint center.

Plotting and editing tools have been developed to facilitate sensitivity studies. The muscle editor lets the user graphically adjust muscle-tendon paths (i.e. origin, insertion

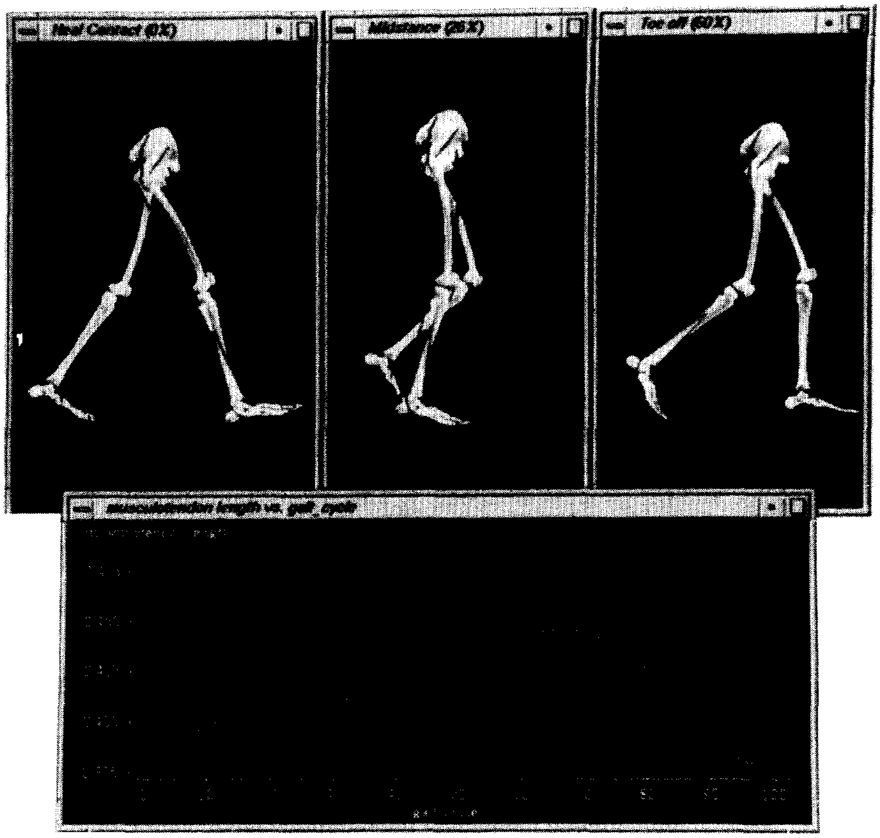
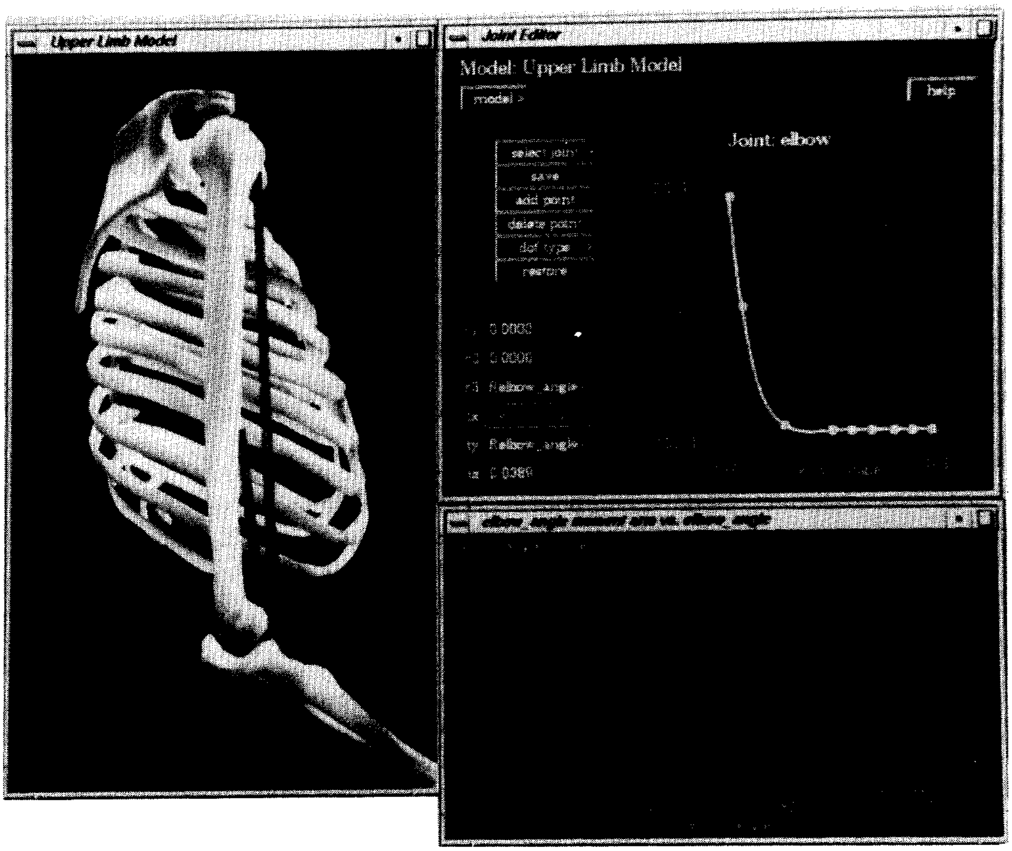


Fig. 4. Animation created from gait analysis data. The windows along the top show a sagittal view of a musculoskeletal model at three points in the gait cycle. Muscle–tendon length (in meters) is plotted for the hamstrings and rectus femoris vs percent of gait cycle.



r1 0.0000
r2 0.0000
r3 f(elbow_angle)
tx
ty f(elbow_angle)
tz 0.0389

Fig. 5. Display from an analysis of elbow mechanics. The kinematics of the elbow can be altered with the joint editor (upper right window) by changing the rotations (*r*1 *r*2 *r*3) and translations (*tx ty tz*) that define the joint. The kinematic function for *tx*, the anterior–posterior translation of the ulna with respect to the humerus, is displayed and can be changed by moving the control points of the cubic spline that define this function. The effects of these changes are analyzed by manipulating the model (left window), and plotting moment arms (lower right window).

and via-point coordinates) and parameters (F_o^M, ℓ_o^M, ℓ_s^T, α_o, V_m) to study the sensitivity of each muscle's moment-generating characteristics to these parameters. For instance, the sensitivity of a muscle's moment-generating capacity to variations in its muscle-tendon path can be studied by altering the path and plotting the change in the maximum isometric joint moment for a range of body positions.

File-saving features enable the user to save musculoskeletal models and analyses of them. Descriptions of altered joints and muscles can be saved in files containing the same information that is included in joint and muscle input files. Plots can be saved in Postscript™ files for printing or editing by other applications. Animations can be created by positioning the model in a desired configuration and then saving this configuration in a file; this can be repeated any number of times to create a sequence of frames that define an animation.

DISCUSSION

Design goals

SIMM was designed to be *general*, *interactive* and *extensible*, and to allow development of *accurate* musculoskeletal models. *Generality* was achieved by making the code completely independent of the particular system to be modeled. For example, all of the menus that contain muscle, joint, or body segment names are formed after the input files have been loaded, so they are not specific to a particular musculoskeletal structure. The partial velocity method computes moment arms for all types of joints. Also, the software can handle any number of muscles and body segments that are connected in arbitrary ways. Thus, the cat neck or hindlimb can be modeled in as straightforward a manner as the human upper and lower limbs.

A set of window-oriented tools allows the user to *interact* with musculoskeletal models by changing any parameter of a model without leaving the program. Once created, a musculoskeletal model can be viewed, manipulated, and analyzed using screen menus, the mouse, and user-defined keys. Computing the effects of an alteration usually takes less than 5 s. Our goal was to create a modeling environment in which a scientist can explore how muscle properties, skeletal geometry, and joint kinematics interact to produce moments about the joints. Since our aim was to make this environment useful to those who may not be experienced in computer programming, we have attempted to make the user interface unambiguous. We have found that new users can learn to manipulate and analyze existing models in approximately 2 h. Learning to define a new model, however, takes considerably longer.

To achieve *extensibility* in the software, we have implemented the editing and analysis tools as a set of distinct, modular components. Thus, little of the existing code needs to be changed to extend the capabilities of the program. Rather, to add a new tool to the program, subroutines that handle the tool's functions can be inserted into the existing software, which would require only minor changes.

Limitations

It is important to discuss the limitations in the *accuracy* of models developed using this software. SIMM allows each muscle-tendon path to be described as a series of line segments. This is reasonable for muscles with small areas of origin and insertion (e.g. tibialis posterior, flexor carpi radialis). However, muscles with large areas of attachment, multiple origins, or curved paths may be more realistically modeled as volumetric objects. As a compromise, muscles with large or multiple attachment areas (e.g. the deltoid and iliopsoas) can be split into any number of compartments. Similarly, a ligament can be separated into a number of different fibers to represent its complex architecture.

The joint modeling technique allows the kinematics of any joint to be characterized. Kinetic joint models, in which joint motion is governed by the forces in muscles, ligaments and other tissues, would provide a more comprehensive representation of joint motion than do predefined kinematic functions. Such kinetic models would also have the advantage of computing joint contact forces, which are not directly available in models developed with SIMM. Nevertheless, to create accurate kinematic models, users must take into account constraints imposed by cartilage, ligaments, and bone shape when specifying kinematic functions. Thus, the effects of these tissues on joint motion are contained in SIMM-based models, albeit implicitly.

Our method of computing muscle and tendon forces is based on a dimensionless muscle-tendon model that can be scaled to represent a wide range of muscle architectures [16]. Although our implementation of the model described by Zajac [16] characterizes the force–length–velocity relationships of muscle, several other properties of muscle (e.g. activation dynamics and yielding) are not currently included. Also, the model assumes that all fibers within a muscle are at the same length for a given muscle-tendon length. This assumption may result in an underestimation of the range of joint angle over which a muscle can produce active force [18]. We imposed this constraint to avoid an additional parameter, such as the standard deviation of muscle-fiber lengths within a given muscle, for which there are little experimental data.

Muscle force calculations also depend on muscle activation. Reasonable estimates of muscle activation can be obtained during steady-state, isometric tasks by processing EMG signals [19]. These activation estimates can be incorporated into a SIMM model to estimate isometric muscle forces and joint moments during a sequence of static postures. In other situations, one may assume that a muscle, or group of muscles, is fully activated to estimate its maximum moment-generating potential [20]. However, reliable estimates of muscle activation patterns during movement are more difficult to obtain. Thus, care should be taken to verify the reliability of the activation data and, if possible, to test the accuracy of the resulting force computations.

Obtaining model parameters

Developing models with SIMM depends on the availability of reliable experimental data needed to determine model parameters. The number of parameters in a model depends on the model's complexity and the type of analysis one desires to perform. To calculate lengths and moment arms, one needs to specify only muscle-tendon paths and joints kinematics.

Muscle-tendon paths have been derived from various anatomical measurements. For example, Brand *et al.* used biplanar radiographs to estimate muscle attachments in the human lower limb [21]. Wood *et al.* used a 3D digitizer to determine the centroidal paths of muscles crossing the shoulder [9]. An *et al.* constructed the paths of muscles crossing the elbow from CT cross-sections [22]. Yamaguchi *et al.* summarized available muscle-path data and provided a table that lists the 3D coordinates of attachment sites for many muscles in the human lower limb, upper limb, torso, head and neck [23]. Muscle attachments have also been measured in other animals, such as the cat [13]. Even though descriptions of muscle attachments exist in the literature, we have found that it is extremely important to display and evaluate these data in a number of body positions. Muscle paths that appear accurate in one body position (e.g. the position in which they were measured) may erroneously pass through bones or other structures in other positions. Without graphical display, these inaccuracies are difficult to detect.

Joint files are based on experimental measurements of joint kinematics. For revolute joints, only the location and orientation of the joint axis must be specified. For more complex joints, such as the human shoulder, one must specify the transformations between body segmental reference frames as functions of the generalized coordinates. Experimental measurements of joint motion that are expressed in this form (e.g. [24]) can be implemented directly. Otherwise, experimental data must be converted into a format that is compatible with SIMM.

Four *muscle-tendon parameters* (peak isometric force, optimal muscle-fiber length, pennation angle and tendon slack length) must be specified if one desires to calculate muscle force as a function of length. These parameters are usually derived in anatomical studies in which physiologic cross-sectional areas (from which peak isometric forces can be derived [20]), muscle-fiber lengths and pennation angles are measured. Several investigators have measured these parameters for the human upper limb [22, 25–27], lower limb [28, 29], and other structures (see [23] for review). Muscle-fiber lengths have also been measured for the hindlimb of the cat [30, 31] and the frog [32]. Very few studies have measured tendon slack length, although Hoy *et al.* have described a method to estimate this parameter [20].

Bone files are not needed to create a model in SIMM because joint kinematics are defined explicitly and are not dependent on the interaction of bone surface models. However, without bone surfaces it is impossible to visualize the geometric relations among the muscles and bones and thus the utility of SIMM is greatly decreased. Accurate bone models can be derived by digitizing bone surfaces or from computed tomographic data [3].

Applications

SIMM is used in a wide variety of applications. One of the most basic functions of the software is to represent musculoskeletal geometry and calculate muscle moment arms. Murray *et al.* developed a computer model of the elbow joint and its surrounding musculature to explore the variation of muscle moment arms with elbow position [33]. The computer model includes three-dimensional representations of the rib cage, scapula, clavicle, humerus, ulna and radius which were obtained by digitizing a male skeleton

(Fig. 5). Elbow flexion/extension is represented as a uniaxial hinge joint with its axis passing through the centers of the capitulum and the trochlear sulcus, two anatomical landmarks that were easily identified on the computer model. Muscle-tendon paths were also defined based on landmarks of the three-dimensional bone models. The long and short heads of the biceps, the brachialis, the brachioradialis, the pronator teres, and the long, medial, and lateral heads of the triceps were each represented as a series of points connected by line segments. Via points and wrapping points were included to represent the path of the muscles that are constrained by retinacula or wrap around a bone. Moment arms were measured in two anatomic specimens to test that accuracy of the model. These comparisons demonstrate that the computer model accurately represents the geometry of the elbow muscles [33].

SIMM has also been used to analyze the biomechanical consequences of surgical procedures. For example, tendon transfer surgeries in the human lower limb have been studied by graphically altering the muscle-tendon paths and quantifying the effects on muscle moment arms [34] and force-generating capacities [35]. This is similar to the approach used by Buford and Thompson to simulate tendon transfers in the hand [10]. However, since these investigators did not calculate muscle-tendon forces, their analyses of simulated tendon transfers are limited to the effects on moment arms and tendon excursions. If their detailed models of the thumb and finger tendons [36] were implemented within SIMM, along with specification of the muscle-tendon parameters [25], analyses could include the effects on muscle-tendon forces and moments. SIMM has also been used to study total hip replacements [37] and bony reconstructions [6].

In clinical motion analysis, well designed three-dimensional animations of human movement can help demonstrate the causes of complex movement disabilities and to communicate findings to others. For example, a computer animation can be used to observe motion from any angle, at any speed. Body segments can be eliminated if they obstruct the view of other segments. Electromyographic activity of the muscles and ground reaction forces can be visualized in the context of the joint kinematics. We have found the animations created with SIMM to be extremely helpful in discovering the causes of movement abnormalities. The software is currently used at several major gait analysis laboratories in this capacity.

Regardless of the application, it is imperative that the accuracy of musculoskeletal models be thoroughly tested. The most basic test is to verify that the musculoskeletal geometry is consistent with anatomical constraints (e.g. muscles do not pass through bones). The kinematic descriptions of the joints must also be tested to assure that the bones do not pass through each other or separate abnormally as the joint is moved through a range of motion. The moment arms estimated based on the musculoskeletal geometry and joint kinematics must then be compared to experimentally measured moment arms [33]. Finally, the joint moments computed with the model can be compared to experimentally measured joint moments [7, 8]. In any case, the responsibility of testing and evaluation is on the user who develops each new musculoskeletal model.

SUMMARY

When developing and analyzing models of musculoskeletal structures, biomechanists usually want to focus on how the model parameters affect muscle and joint function rather than on writing software to implement the model on a computer. SIMM is the first interactive, graphics-based software package that enables users to develop models of many different musculoskeletal structures without programming. There are a number of advantages of creating models with SIMM. First, it provides a framework within which model parameters can be organized. Model complexity can be increased one step at a time and the effect of each added complexity can be critically evaluated. Once a model is developed, it is straightforward to transfer the model to another research group, where it can be further tested and used. The software is used at approximately 25 centers worldwide. Our goal is to provide a framework that will allow investigators to create a library of musculoskeletal models that can be exchanged, tested, and improved through a multi-institutional collaboration.

Acknowledgements—We gratefully acknowledge Eric Topp for suggesting the partial velocity method of computing moment arms. We also thank Melissa Gross, Eric Topp and Felix Zajac for the many discussions we have had related to musculoskeletal modeling and Idd Delp for help with the illustrations. We are grateful to Wendy Murray and Thomas Buchanan for their assistance with the elbow model shown in Fig. 5. The development of SIMM was supported by the Rehabilitation Research and Development Service of the Department of Veterans Affairs.

REFERENCES

1. R. A. Robb, D. P. Hanson, R. A. Karwoski, A. G. Larson, E. L. Workman and M. C. Stacy, ANALYZE: a comprehensive, operator-interactive software package for multidimensional medical image display and analysis, *Comput. Med. Imaging Graphics* **13**, 433–454 (1989).
2. E. J. Farrell and R. A. Zappulla, Three-dimensional data visualization and biomedical applications, *CRC Crit. Rev. Biomed. Eng.* **16**, 323–363 (1989).
3. M. W. Vannier, W. G. Totty, W. G. Stevens, P. M. Weeks, D. M. Dye, V. J. Daum, L. A. Gilula, W. A. Murphy and R. H. Knapp, Musculoskeletal applications of three-dimensional surface reconstructions, *Orthop. Clin. North. Am.* **16**, 543–555 (1985).
4. J. W. Granholm, D. D. Robertson, P. S. Walker and P. C. Nelson, Computer design of custom femoral stem prostheses, *IEEE Comput. Graphics Appl.* **7**, 26–35 (1987).
5. S. B. Murphy, P. K. Kijewski, M. B. Millis, J. E. Hall, S. R. Simon and H. P. Chandler, The planning of orthopaedic reconstructive surgery using computer-aided simulation and design, *Comput. Med. Imaging Graphics* **12**, 33–45 (1988).
6. S. L. Delp, E. E. Bleck, F. E. Zajac and G. Bollini, Biomechanical analysis of the Chiari pelvic osteotomy, *Clin. Orthop. Rel. Res.* **254**, 189–198 (1990).
7. S. L. Delp and W. J. Maloney, Effects of hip center location on the moment-generating capacity of the muscles, *J. Biomech.* **26**, 485–499 (1993).
8. S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp and J. M. Rosen, An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures, *IEEE Trans. Biomed. Eng.* **37**, 757–767 (1990).
9. J. E. Wood, S. G. Meek and S. C. Jacobsen, Quantification of human shoulder anatomy for prosthetic arm control-I. Surface modelling, *J. Biomech.* **22**, 273–292 (1989).
10. W. L. Buford and D. E. Thompson, A system for three-dimensional interactive simulation of hand biomechanics, *IEEE Trans. Biomed. Eng.* **BME-34**, 444–453 (1987).
11. T. S. Buchanan, M. J. Moniz, J. P. A. Dewald and W. Z. Rymer, A biomechancial model of muscle forces about the wrist joint during isometric tasks, *J. Biomech.* **26**, 547–560 (1993).
12. M. T. Mai and R. L. Lieber, A model of semitendinosus muscle sarcomere length, knee and hip joint interaction in the frog hindlimb, *J. Biomech.* **23**, 271–279 (1990).
13. C. R. Wickland, J. F. Baker and B. W. Peterson, Torque vectors of neck muscles in the cat, *Exp. Brain. Res.* **84**, 649–659 (1991).
14. E. Y. Chao and B. F. Morrey, Three-dimensional rotation of the elbow, *J. Biomech.* **11**, 57–73 (1978).
15. G. E. Forsythe and M. A. Malcolm, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs (1977).
16. F. E. Zajac, Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control, *CRC Crit. Rev. Biomed. Eng.* **17**, 359–411 (1989).
17. K. N. An, K. Takakashi, T. P. Harrigan and E. Y. Chao, Determination of muscle orientation and moment arms, *J. Biomech. Eng.* **106**, 280–282 (1984).
18. W. Herzog and H. E. D. J. Ter Keurs, Force-length relation of *in-vivo* human rectus femoris muscles, *J. Physiol. (Europ).* **411**, 642–647 (1988).
19. F. E. Zajac and M. E. Gordon, Determining muscle's force and action in multi-articular movement, *Exercise and Sport Sciences Reviews* **17**, 187–230 (1989).
20. 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. Biomech.* **23**, 157–169 (1990).
21. R. A. Brand, R. D. Crowninshield, C. E. Wittstock, D. R. Pederson, C. R. Clark and F. M. van Krieken, A model of lower extremity muscular anatomy, *J. Biomech. Eng* **104**, 304–310 (1982).
22. K. N. An, F. C. Hui, B. F. Morrey, R. L. Linchield and E. Y. Chao, Muscles across the elbow joint: a biomechanical analysis, *J. Biomech.* **14**, 659–669 (1981).
23. G. T. Yamaguchi, A. G. U. Sawa, D. W. Moran, M. J. Fessler and J. M. Winters, A survey of human musculotendon actuator parameters, *Multiple Muscle Systems*, pp. 717–773. Springer-Verlag, New York (1990).
24. C. Hogfors, B. Peterson, G. Sigholm and P. Herberts, Biomechanical model of the human shoulder joint—II. The shoulder rhythm, *J. Biomech.* **24**, 699–709 (1991).
25. P. W. Brand, R. B. Beach and D. E. Thompson, Relative tension and potential excursion of muscles in the forearm and hand, *J. Hand Surg.* **6**, 209–219 (1981).
26. R. L. Lieber, M. F. Babak and M. J. Botte, Architecture of selected wrist flexor and extensor muscles, *J. Hand Surg.* **15A**, 244–250 (1990).
27. R. W. Bassett, A. O. Browne, B. F. Morrey and K. N. An, Glenohumeral muscle force and moment mechanics in a position of shoulder instability, *J. Biomech.* **23**, 405–415 (1990).
28. J. A. Friederich and R. A. Brand, Muscle fiber architecture in the human lower limb, *J. Biomech.* **23**, 91–95 (1990).

29. T. L. Wickiewicz, R. R. Roy, P. L. Powell and V. R. Edgerton, Muscle architecture of the human lower limb. *Clin. Orthop.* **179**, 275–283 (1983).
30. P. M. Rack and D. R. Westbury, Elastic properties of the cat soleus tendon and their functional importance, *J. Physiol. (London)* **347**, 479–495 (1984).
31. R. D. Sacks and R. R. Roy, Architecture of the hind limb muscles of cats: functional significance, *J. Morphol.* **173**, 185 (1983).
32. R. L. Lieber and J. L. Boakes, Sarcomere length and joint kinematics during torque production in the frog hindlimb, *American Journal of Physiology* **254**, 759–768 (1988).
33. W. A. Murray, S. L. Delp and T. S. Buchanan, Variation of muscle moment arms with elbow and forearm position, *J. Biomech* **28**, 513–525 (1994).
34. S. L. Delp, D. Ringwelski and N. C. Carroll, Transfer of the rectus femoris: effects of transfer site on moment arms about the knee and hip, *J. Biomech.* **27**, 1201–1211 (1994).
35. S. L. Delp, J. P. Loan, M. G. Hoy and J. M. Rosen, Computer simulation of lower-extremity tendon transfers, *Proc. 36th Annual Meeting, Orthopaedic Research Society*, p. 537, (1990).
36. D. E. Thompson and D. J. Giurintano, A kinematic model of the flexor tendons of the hand, *J. Biomech.* **22**, 327–334 (1989).
37. S. L. Delp, A. V. Komattu and R. L. Wixson, Superior displacement of the hip in total joint replacement: effects of prosthetic neck length, neck-stem angle, and anteversion angle on the moment-generating capacity of the muscles, *J. Orthop. Res.* **12**, 860–870 (1994).
38. T. R. Kane and D. A. Levinson, *Dynamics: Theory and Applications*, New York, McGraw-Hill (1985).

About the Author—SCOTT DELP received the B.S. degree in mechanical engineering from Colorado State University in 1983, and the M.S. and Ph.D. degrees in mechanical engineering from Stanford University in 1986 and 1990, respectively. He worked at Hewlett Packard's Graphics Workstation Division from 1983 to 1985 and the Veterans Affairs Rehabilitation Research and Development Center in Palo Alto, CA from 1986 to 1990. Since 1991, he has been an Assistant Professor of Biomedical Engineering and Rehabilitation Medicine at Northwestern University and a Senior Research Scientist at the Rehabilitation Institute of Chicago.

Dr Delp's research focuses on biomechanics of human movement, with emphasis on graphics-based simulation of surgeries and dynamics-based analysis of movement. He was awarded a Baxter Faculty Fellowship in 1991 and a National Science Foundation Young Investigator Award in 1992. He is a member of the American Society of Mechanical Engineers, the American Society of Biomechanics, the International Society of Biomechanics, and the Orthopaedic Research Society.

About the Author—PETER LOAN received B.S. degrees in mechanical engineering and philosophy from the Massachusetts Institute of Technology in 1986, and M.S. degrees in mechanical engineering and computer science from Stanford University in 1989. He worked at the Veterans Affairs Rehabilitation Research and Development Center in Palo Alto, CA from 1986 to 1992. Since 1992, Mr Loan has served as Director of Research and Development at MusculoGraphics, Inc.

Mr Loan's research and engineering efforts have concentrated on the biomechanics of the human musculoskeletal system. He has extensive experience developing computer graphics simulations of the human body, and has created several simulation programs that are used by researchers worldwide to gain insight into the function of the musculoskeletal system.