Virtual muscle: a computational approach to understanding the effects of muscle properties on motor control

Ernest J. Cheng [1], Ian E. Brown, Gerald E. Loeb *

Department of Biomedical Engineering, DRB-B12, Medical Device Development Facility, Alfred E. Mann Institute for Biomedical Engineering, University of Southern California, MC 1112, 1042 West 36th Place, Los Angeles, CA 90089-1112, USA

Abstract

This paper describes a computational approach to modeling the complex mechanical properties of muscles and tendons under physiological conditions of recruitment and kinematics. It is embodied as a software package for use with Matlab™ and Simulink that allows the creation of realistic musculotendon elements for use in motor control simulations. The software employs graphic user interfaces (GUI) and dynamic data exchange (DDE) to facilitate building custom muscle model blocks and linking them to kinetic analyses of complete musculoskeletal systems. It is scalable in complexity and accuracy. The model is based on recently published data on muscle and tendon properties measured in feline slow- and fast-twitch muscle, and incorporates a novel approach to simulating recruitment and frequency modulation of different fiber-types in mixed muscles. This software is distributed freely over the Internet at http://ami.usc.edu/mddf/virtualmuscle. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Matlab; Simulink; Neurophysiology; Biomechanical modeling; Muscle; Simulation

1. Introduction

The study of motor control seeks to determine how the central nervous system effects functional, goal-directed movement. For conceptual purposes, it is useful to visualize the mechanisms involved in motor control as a hierarchy, with the brain at the top, the spinal cord in the middle, and the musculoskeletal plant at the bottom (Loeb et al., 1999). Researchers typically have attempted to understand the mechanisms of motor control by tackling one level of the hierarchy at a time. For example, the 'population vector' paradigm (reviewed in Georgopoulos, 1986), where directional control of hand movements is correlated with cortical activity in direction-specific neurons, is concerned mostly with the top level of the hierarchy, the brain. Other theories of motor control have emphasized the role of reflexes at the middle, spinal level of the hier-

archy in setting 'equilibrium points' as targets for movements (e.g. Asatryan and Feldman, 1965). However, it has been realized that many discrepancies between the proposed theories and observed behavior might be explained by the fact that the actual effectors of movement, the muscles themselves at the lowest level of the hierarchy, have sophisticated non-linear viscoelastic properties (Hogan, 1985; Gottlieb, 1996; Karniel and Inbar, 1997; Todorov, 2000).

Recently, increasingly complex models have attempted a synthesis of many levels of this hierarchy. As more studies use this approach, there is an increasing need for user-accessible models that accurately describe the behavior of muscle under realistic physiological conditions. For example, studies of joint stiffness and static and dynamic muscle function based on optimization criteria are often validated solely with electromyograms (EMG; e.g. Kaufman et al., 1991; van der Helm, 1994; Osu and Gomi, 1999); with a musculoskeletal model, force and impedance could be predicted from EMG and compared with the results of inverse dynamic analyses. Other studies have included viscoelastic muscle elements, but researchers are often required to develop their own musculotendon models (e.g. Karniel

* Corresponding author. Tel.: +1-213-8211112; fax: +1-213-8211120.

E-mail address: gloeb@usc.edu (G.E. Loeb).

[1] Present address: Sapient Corporation; 200 West Adams, Ste. 2700, Chicago, IL 60610, USA

and Inbar, 1997; Gonzalez et al., 1997; Soechting and Flanders, 1997); consequently, these models are usually simplified and designed with arbitrarily estimated parameters.

Our goal was to provide a general model of muscle that would be easy-to-use, easily integrated with models and simulations of kinetics and control, and at the same time accurate and scalable in its level of sophistication. This would obviate the need for researchers to develop their own models of muscle. We integrated several recent models of the recruitment of motor units (Brown, 1998), the contractile properties of mammalian muscle (Brown and Loeb, 2000; Brown et al., 1996b, 1999), and the elastic properties of tendon and aponeurosis (Scott and Loeb, 1995). We have created a graphical user interface (GUI) based software package called Virtual Muscle that allows researchers who may not be interested in muscle or motoneuron physiology per se to generate musculotendon simulation 'blocks' that can be used readily with their existing simulations. This is facilitated by building the model from normalized parameters (Zajac, 1989) that are readily scaled to the morphometry of specific muscles. For the majority of users who need only functional musculotendon elements that accurately reflect the non-linear viscoelastic behavior of muscle, we have provided data on mammalian muscle fiber types that can be combined with the commonly available morphometry of specific muscles (for a summary, see Yamaguchi et al., 1990). For more ambitious users, entirely new fiber types can be created, with user-specified properties and recruitment patterns.

2. Software/hardware requirements

The Virtual Muscle modeling package is designed to run on Matlab 5.2 or higher (The Mathworks Inc., Natick, MA) and also requires the Simulink 2.2 or higher module for Matlab. While the GUI for specifying the parameters for the muscle model runs in Matlab, the individual muscle blocks created for use in the actual simulations are stand-alone Simulink blocks. We chose to develop the muscle model for the Matlab/Simulink combination for several reasons. First, Matlab and Simulink are already widely used in sensorimotor research, especially in developing motor control simulations and studying neural networks. Second, Matlab is available on a wide variety of platforms, including PC compatibles, Apple Macintosh, and various UNIX-based systems. Third, Simulink's block-oriented nature makes it simpler for users to conceptualize the relationships between the various equations that comprise the muscle model. Finally, Matlab is capable of interfacing with a large number of external applications, such as kinetic analysis software to model the segmental dynamics. We have provided a sample implementation of this using the dynamic data exchange (DDE) interface common to many Microsoft Windows based applications. The muscle model package will run on any hardware and under any operating system that is capable of running the required Matlab and Simulink versions.

3. Design of the muscle model

Virtual Muscle was created for use in the context of a hierarchical model of motor control. Skeletal dynamics comprise the lowest level, and are driven by the realistic muscle properties provided by Virtual Muscle at the middle level (see Fig. 1). At the top level, the muscles can be controlled by any arbitrary set of activation commands, ranging from pre-recorded EMG data to dynamic, feedback-driven reflex models, to high level simulations of cortical commands. With the use of two GUI-based Matlab functions, BuildFiberTypes and BuildMuscles, the user defines the properties of each fiber type and the morphometry of each musculotendon element. After the properties of each musculotendon element have been specified, stand-alone Virtual Muscle blocks for use in SIMULINK can be created and integrated with the skeletal dynamics and control systems, which are provided by the user.
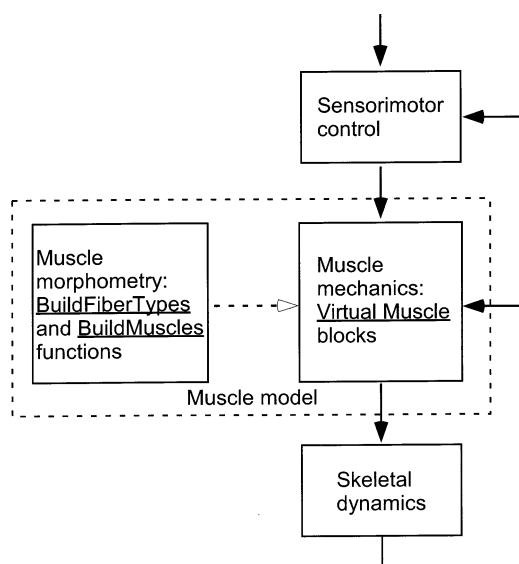

Fig. 1. The interaction between various levels of a hierarchical model of neuromusculoskeletal control. The highest level, sensorimotor control, drives the muscle mechanics level, which in turn drives the skeletal dynamics level. Feedback also exists between each level of the hierarchy. Behavior of the muscle mechanics level is computationally modeled with Virtual Muscle blocks. The fiber- and muscle-specific parameters are specified in the BuildFiberTypes and BuildMuscles functions prior to simulation. These two functions are not active during the simulation, as indicated by the dashed arrow.
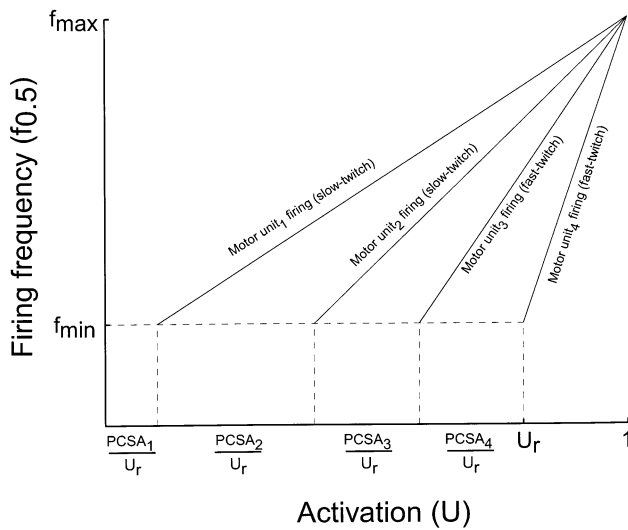
Fig. 2. Recruitment scheme as applied to a hypothetical muscle consisting of two simulated slow-twitch and two fast-twitch motor units. As activation increases, motor units are recruited sequentially, first on the basis of the recruitment rank of the fiber-type (slow-twitch fibers have a lower recruitment rank than fast-twitch), and then by the user-specified order of the motor units. All motor units are recruited when activation reaches U_r, which is user-specified for each muscle. The firing frequency of each motor unit modulates linearly between minimal (f_{min}) and maximal (f_{max}) levels. These levels are defined separately and expressed as multiples of $f_{0.5}$ for each fiber type; this allows slow-twitch motor units to operate over a different range of actual frequencies than fast-twitch units.

The main objective guiding the design of this package was the need to balance simplicity and accuracy. Each Virtual Muscle musculotendon element requires an activation and a length signal as input and returns force as an output, the minimum number of state variables for most models. We have included the following features, which we hope will meet the needs of most researchers.

- A single activation input (e.g. net synaptic drive or EMG envelope) drives a function which serves to recruit incrementally and modulate the firing rate of any number of specified motor units, with the user able to specify the recruitment order and frequency range (Fig. 2).
- Type-specific contractile elements that produce force as a function of recruitment, frequency modulation, length and velocity, including viscoelastic elements for passive muscle force (Fig. 3).
- Passive elastic elements for series-compliance of tendons and aponeuroses.

While the behavior of each of these components can be specified in detail by the user, physiologically realistic default values have been provided for recruitment parameters and for the behavior of commonly modeled mammalian fiber types. Further detail on the design of these three components is provided below.

The recruitment model is based on that described by Brown (1998). Physiologically, muscles are organized into motor units, each consisting of a motor neuron and the several hundred muscle fibers it controls. Groups of similar motor unit types tend to be recruited together, while different motor unit types tend to be recruited in a fixed order. This allows us to simplify the model by lumping many motor units into single simulated motor units; this is justified by the finding that contractile element behavior scales well from the sarcomere level to the whole fiber level and also to the level of the entire recruitment group (Zajac, 1989). In the simplest case, a musculotendon element can be modeled as a single motor unit composed of homogenous muscle fibers; the activation signal then only modulates the firing frequency of this motor unit. If multiple motor units and/or different fiber types are used in each musculotendon element, a single activation input sequentially recruits the motor units based on two properties, first, the recruitment rank of the fiber type for that motor unit defined in BuildFiberTypes, and then secondarily, the order determined in the Build-Muscles function. As U increases, more motor units are recruited until a value called U_r is reached (specified for each muscle in BuildMuscles). U_r is the point at which all motor units have been recruited; increases in activation beyond this point result only in frequency modulation of motor units. Motor unit recruitment threshold is based on the cumulative fractional physiological cross-sectional area (PCSA) of all motor units recruited prior to the given motor unit, with range of recruitment between 0 and U_r (Fig. 2).

In the example illustrated in Fig. 2, the first two compartments are slow-twitch (and are thus recruited first), and the U_r for the muscle is 0.8. Note that while it appears that both slow- and fast-twitch motor units have the same firing frequency range, f_{min} and f_{max} are defined in frequency units of $f_{0.5}$. The normalization factor $f_{0.5}$ is defined as the stimulus frequency necessary to produce 50% of maximal, isometric force (Brown et al., 1999) and thus has different values depending on fiber type. For each motor unit, the frequency modulates from a predetermined f_{min}, when the unit is first recruited, up to f_{max}, which occurs at full activation for the muscle. A linear scaling of firing frequency relative to EMG has been demonstrated experimentally by Milner-Brown et al. (1973a). A common initial firing frequency for motor units of a given fiber type and their convergence to type-specific maximal firing frequency at maximal activation have been demonstrated by De Luca et al. (1996).

To simplify the model, whole muscles are treated as linear combinations of pure slow- and fast-twitch muscle compartments (Zajac, 1989; Brown et al., 1996b). This muscle model differs from almost all others currently available, which have assumed that the contractile properties of muscle scale linearly with neural activation (e.g. Otten, 1987; Durfee and Palmer, 1994;

Shue et al., 1995; Brown et al., 1996b). That assumption contradicts experimental data that suggests otherwise (Rack and Westbury, 1969; Balnave and Allen, 1996). Briefly, various relationships including force–length, force–velocity, activation rise- and fall-times, sag and yield tend to be qualitatively different in fast- and slow-twitch muscle and/or to depend on the frequency of firing of motor units (Rack and Westbury, 1969; Scott et al., 1996; Brown, 1998; Brown and Loeb, 2000).

The in-series compliance of tendons plus aponeuroses has also been included in the model (see Scott and Loeb, 1995). Tendon compliance is important because it causes the length and velocity of the contractile element to be out of phase with those of the whole muscle. For example, in an active muscle with a high ratio of tendon-to-contractile element length, a stretch to the whole musculotendon system would only slightly lengthen the contractile-element, with the majority of the length change being accommodated by lengthening of the tendon (Zajac, 1989). In fact, under certain conditions, the musculotendon element can be lengthening while the contractile element within the muscle is shortening, or vice-versa. This behavior would be misrepresented in tendon-less models, which assume that contractile length changes are proportional to path length changes. Because tendon and aponeurosis have been shown to have similar elastic properties, they have been grouped into a single term for each musculotendon element (Scott and Loeb, 1995). The tendon and the contractile muscle fiber elements both act on a muscle mass that provides inertial damping to prevent computational instabilities from arising within the muscle model (Loeb et al., 1999).

4. Implementation of muscle model

Zajac (1989) advocated a model that defines the properties of a generic musculotendon element with a single set of equations. This approach was based on the assumption that the properties of individual sarcom-
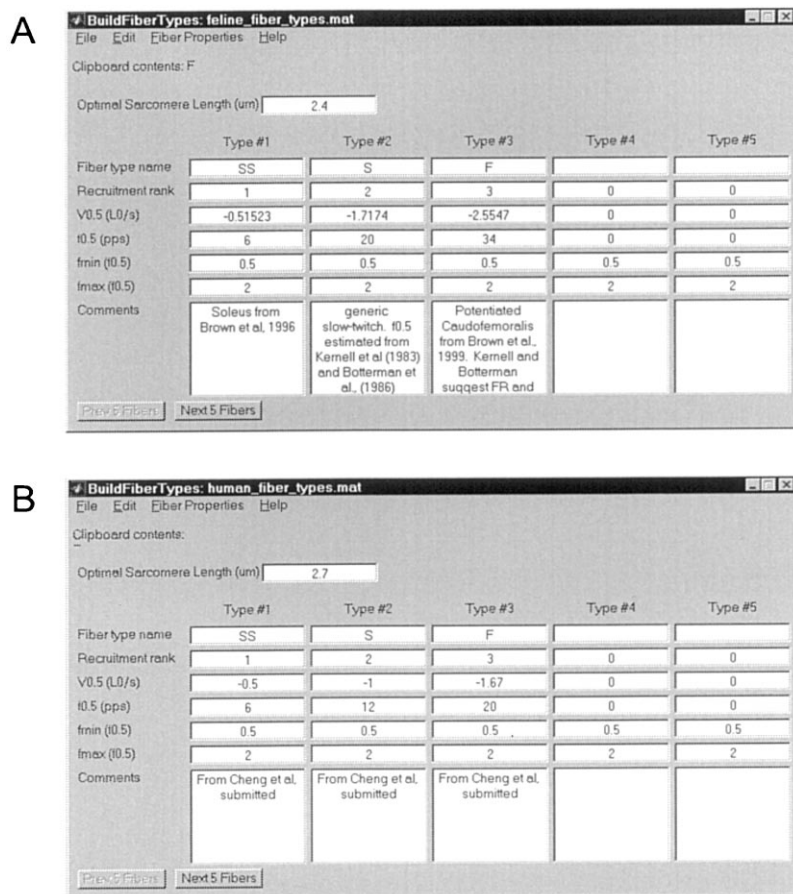

Fig. 3. 'Screen shots' of the BuildFiberTypes function showing the actual feline and human fiber type databases provided with Virtual Muscle. Each fiber type is entered into a different column. Modifying the parameters in this window can affect the recruitment behavior, active and passive force–length relationships, force–velocity relationships and rise and fall times. We created the human fiber type database by modifying these parameters, as described in Appendix B. The individual coefficients used in the muscle model (see Fig. 4) can be modified directly for each fiber type in a second window available under this GUI (not shown).

eres, when grouped together, can be scaled to represent the properties of muscle fibers and these, in turn, can be grouped to represent whole muscle properties. The five parameters required by Zajac's equations were L_0^M (optimal fascicle length, the length at which the muscle produces maximal tetanic isometric force), F_0^M (maximal tetanic isometric force), α_0 (pennation angle of the muscle fibers at L_0^M), τ_C (time-scaling parameter for maximal muscle shortening velocity and rise and fall times), and L_S^T (tendon slack length). Once these measures had been determined for a given muscle, its behavior could be reproduced mathematically.

4.1. Defining muscle fiber types

Virtual Muscle utilizes a similar approach in scaling each motor unit in the simulation, with the following modifications (Brown and Loeb, 2000; Brown et al., 1999). In our model we currently ignore pennation angle. Zajac's fiber-specific parameter τ_C has been replaced with $V_{0.5}$ (velocity of shortening required to produce 0.5 F_0 at 1.0 L_0 during a tetanic stimulus; used to scale the force–velocity dependencies) and $f_{0.5}$ (stimulus frequency necessary to produce 0.5 F_0 at 1.0 L_0 during a tetanic, isometric contraction; used to scale rise and fall times). Although $V_{0.5}$ and $f_{0.5}$ appear to scale proportionally with each other in normal muscle (see Fig. 8), some of the mechanisms underlying each parameter are independent. In abnormal muscles, such as those found in paralyzed patients, this relationship may not hold, so the user can specify them separately. The reasons for choosing $V_{0.5}$ for scaling velocity dependencies over the more widely cited V_{max} are two-fold, (i) because the experimental force–velocity plots are nearly asymptotic, it is much easier to measure $V_{0.5}$ accurately than it is to extrapolate V_{max}; and (ii) under physiological conditions, muscles are more likely to operate over a range including $V_{0.5}$ than V_{max}. The reasons for using $f_{0.5}$ to scale the frequency dependency of force production are analogous and have been discussed previously (Brown et al., 1999).

The BuildFiberTypes function allows the user to modify a fiber type database file or to create an entirely new one. If an existing fiber type database exists, however, then researchers not interested in the details of muscle fiber physiology may never need to use this function. This function has two levels of detail. In the first level (Fig. 3), the database has general parameters applicable to all fiber types, such as optimal sarcomere length, as well as general parameters for each individual fiber type, such as name, recruitment rank, etc. If modifications are being made to an existing database, three parameters can be modified in this first level to match measurements from other studies in order to automatically scale some of the basic fiber type properties. Changing optimal sarcomere length can be used to

automatically scale the active and passive force–length properties of all fiber types in the database (by accepting this option in a dialog box). This scaling is based on the assumption that the thick filament length is constant for all fiber types of a given species (Herzog et al., 1992). Changing the fiber type-specific $V_{0.5}$ or $f_{0.5}$ can automatically scale the force–velocity relationship and rise- and fall-times, respectively. We assume that the shortening and lengthening half of the *FV* relationships scale proportionally with $V_{0.5}$ and to each other, and that rise and fall times scale proportionally to $f_{0.5}$. Additionally, if the user wishes to assume that the $V_{0.5}$ and $f_{0.5}$ parameters scale with each other (see Appendix B) then changing one can automatically scale the other. ('Help' screens in the BuildFiberTypes function provide more information regarding the details of these scaling techniques.) Two additional terms on this first level define the minimal and maximal firing frequency for the fiber type, f_{min} and f_{max}, respectively (which default to 0.5 and 2.0, respectively, in units of $f_{0.5}$) and are used by the BuildMuscles function. We show two sample databases in Fig. 3 which we provide with Virtual Muscle. The first, a feline database, is based upon previously published parameters. The second, a human fiber type database, was extrapolated from the feline database using the options just described (see Appendix B for details).

The functions for each fiber type or tendon property require various coefficients that can be inspected and modified in the second level of BuildFiberTypes. This level is less likely to be used than the first. The nature and dependencies of these functions are shown schematically in Fig. 4. Each balloon in the figure corresponds to an equation used in the computation of the force generated by the muscle; these equations and their coefficients for feline muscle have been provided previously (Brown and Loeb, 2000; Brown et al., 1999). We provide the complete set of equations here in Table 1, with a modified passive force–length equation (see Appendix A), the addition of a small passive viscosity for stability purposes during simulation (see Appendix A), and coefficients for human fiber types (see Appendix B). These equations were chosen to correspond to specific physiological mechanisms that dictate the contractile properties of muscle. For example, the well-documented force–length relationship in muscle is known to correspond at a molecular level to the degree of overlap between actin and myosin filaments within the muscle. While this technique of characterizing each relationship in a separate equation may increase the apparent complexity of the model, it has two advantages. First, it allows users to see how the various processes within the muscle interact as opposed to obscuring them with a single, all-encompassing equation. Second, it is readily adaptable to represent muscle properties measured in different fiber types or animal
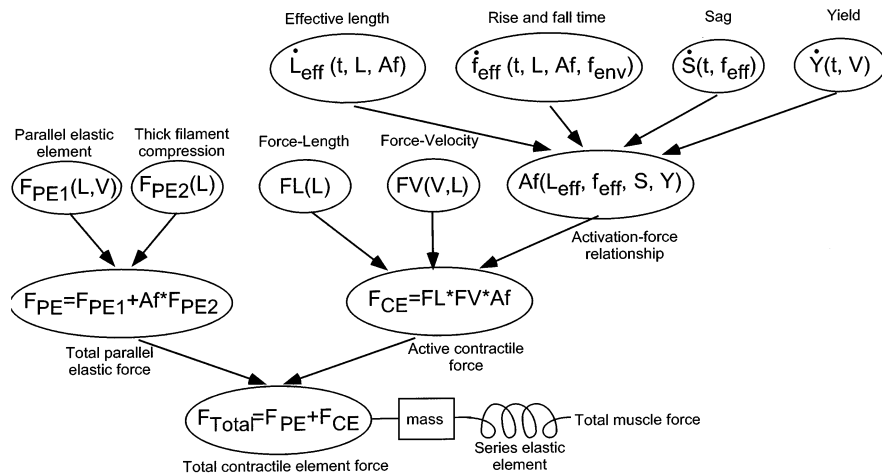
Fig. 4. Schematic representation of the model equations and terms. These elements were designed to have a one-to-one correspondence with the physiological substrates of muscle contraction. The behavior of each element is governed by an equation driven by one to four input variables, with one to seven user-modifiable coefficients. The coefficients can be modified in the BuildFiberTypes function. Complete descriptions of these elements can be found in Brown and Loeb (2000), Brown et al. (1999). F_{PE1} represents the passive visco-elastic properties of stretching a muscle. F_{PE2} represents the passive resistance to compression of the thick filaments at short muscle lengths. *FL* represents the tetanic, isometric force–length relationship. *FV* represents the tetanic force–velocity (*FV*) relationship. *Af* represents the isometric, activation–frequency (*Af*) relationship. f_{eff} represents the time lag between changes in firing frequency and internal activation (i.e. rise and fall times). L_{eff} represents the time lag between changes in length and the effect of length on the *Af* relationship. *S* represents the effects of 'sag' on the activation during a constant stimulus frequency. *Y* represents the effects of yielding (on activation) following movement during sub-maximal activation.

species. It also improves the likelihood that the complete model will be reasonably accurate and computationally well-behaved when operated under combinations of specific morphometry and input conditions that were not studied experimentally in the source data.

There are three other 'global' properties that affect all the fiber types defined within a single fiber type database. The specific tension, or force per cm^2 of PCSA has been shown to be largely independent of histochemical fiber type (Spector et al., 1980; Lucas et al., 1987; Brown et al., 1998); we use 31.8 N cm^{-2} as an overridable default (Scott et al., 1996; Brown et al., 1998). Specific compliance of series tendon and aponeurosis (scaled to PCSA and L_0^T) has a nonlinear form based on feline soleus (Scott and Loeb, 1995). We also include a passive viscosity for the fascicles with a very small default value to provide stability during simulations.

Useful functions such as cutting, copying and pasting fiber type parameters along with the underlying coefficients from one column to another are supported. Fiber types can be imported from one database to another.

4.2. Defining muscle morphometry

The terms required for our modified Zajac-type model that are independent of fiber type and are muscle-specific are L_0, F_0, L_0^T and L_{max}. The first two are the same as L_0^M and F_0^M defined above and by Zajac (we drop the 'M' so as to not imply muscle, which

many people interpret as including tendon and/or aponeurosis — L_0 and F_0 are specific to the muscle fascicles). L_0^T (length of the tendon at maximal tetanic isometric force) replaces Zajac's tendon slack length L_S^T. L_S^T is less well defined that L_0^T and tends to be about 5% shorter. Passive tension is based on the new term L_{max}, which is the maximal length of the fascicles at the maximal anatomical length of the muscle, following the observation that passive force is more closely correlated with L_{max} than with L_0 (Brown et al., 1996a; see Appendix A for more details). These terms are specified for each muscle in the second Matlab function, called BuildMuscles. Once these parameters have been specified, the user can generate blocks that encapsulate the behavior of the entire musculotendon element. The fiber types that comprise the muscles are drawn from a fiber type database file generated in BuildFiberTypes or pre-supplied by us (we currently offer both feline and human fiber type databases); the BuildMuscles function will not work unless an existing fiber type database file is specified. For each musculotendon element to be modeled, the user inputs a single row of data into the main BuildMuscles GUI window (Fig. 5A, left). Some data columns are blank and require user input. The required morphometric values are muscle name, mass, optimal fascicle length, optimal tendon length, and the maximal anatomical musculotendon path length. These morphometric measures are either used directly or converted into the appropriate parameters as required by the set of equations described previously. For example, the L_0 term corre-

sponds directly to the value entered for fascicle length. On the other hand, F_0 is proportional to the physiological cross-sectional area (PCSA) of the muscle; this is derived by the BuildMuscles function from the specified muscle mass (which is converted to volume) and fascicle length. Physiological cross-sectional area (PCSA), (fascicle) L_{max} and F_0 are presented in a column, which is 'grayed out' and cannot be modified directly. Values in these columns are shown only as information for the user, and can be modified only by changing the inputs to their derivation.

Once the morphometry of the muscle has been entered, the user specifies the desired proportion of the muscle that is to be allocated to each muscle fiber type, using fiber type names defined previously in the BuildFiberTypes function. The user specifies the number of sequentially recruited motor units that should be used to simulate each fiber type in the muscle. In this example, we have created a muscle consisting of 60% slow- and 40% fast-twitch muscle fibers, with five motor units allocated to the slow-twitch portion and two motor units to the fast-twitch portion of the muscle (Fig. 5A, right). The recruitment threshold of the motor units in the muscle are controlled in part by the muscle-specific value entered for U_r (see Fig. 2 and design for explanation).

When muscle fiber types are first allocated to portions of the muscle, BuildMuscles automatically creates individual motor units with the earlier recruited units having a smaller size than the later recruited units. The purpose of this is two-fold, first, to accord with the Henneman (1968) size principle of motor unit recruitment, and second, to smooth the force profile at low activation as additional motor units are recruited. If the user is not satisfied with this scheme of motor unit size allocation, it is possible to adjust manually the size of each motor unit individually (Fig. 5B).

As with the BuildFiberTypes function, new muscles can be built by cutting, copying, pasting and editing individual muscles from one row to another and from other databases. Once the muscle model database is completed, it can be saved to a file for future use or modification. To create models of the muscles described in the database, the BuildMuscles function generates Simulink blocks as described in the following section.

Table 1
Summary of model equations and best-fit constants

| Curve	Typical slow twitch fibers				Fast-twitch fibers			
$F_{SE}(L^T) = c^T k^T \ln\left\{\exp\left[\dfrac{(L^T - L_r^T)}{k^T}\right] + 1\right\}$	c^T 27.8	k^T 0.0047	L_r^T 0.964		c^T (same as slow twitch)	k^T	L_r^T	
$F_{PE1}(L) = c_1 k_1 \ln\left\{\exp\left[\dfrac{(L/L_{max} - L_{r1})}{k_1}\right] + 1\right\} + \eta V$	c_1 23.0	k_1 0.046	L_{r1} 1.17	η 0.001	c_1 (same as slow twitch)	k_2	L_{r2}	
$F_{PE2}(L) = c_2\{\exp[k_2(L - L_{r2})] - 1\}, \quad F_{PE2} \leq 0$	c_2 −0.020	k_2 −21.0	L_{r2} 0.70		c_2 (same as slow twitch)	k_2	L_{r2}	
$FL(L) = \exp\left(-abs\left\|\dfrac{L^B - 1}{\omega}\right\|^\rho\right)$	ω 1.12	β 2.30	ρ 1.62		ω 0.75	β 1.55	ρ 2.12	
$FV(V, L) = \begin{cases}(V_{max} - V)/(V_{max} \pm (c_{v0} + c_{v1}L)V), & V \leq 0 \\ (b_v - (a_{v0} + a_{v1}L + a_{v2}L^2)V)/(b_v + V), & V > 0\end{cases}$	V_{max} −7.88	c_{v0} 5.88	c_{v1} 0		V_{max} −9.15	c_{v0} −5.70	c_{v1} 9.18	
	a_{v0} −4.70	a_{v1} 8.41	a_{v2} −5.34	b_v 0.35	a_{v0} −1.53	a_{v1} 0	a_{v2} 0	
$Af(f_{eff}, L_{eff}, Y, S) = 1 - \exp\left[-\left(\dfrac{YSf_{eff}}{a_f n_f}\right)^{n_f}\right], \quad n_f = n_{f0} + n_{f1}\left(\dfrac{1}{L_{eff}} - 1\right)$	a_f 0.56	n_{f0} 2.1	n_{f1} 5		a_f 0.56	n_{f0} 2.1	n_{f1} 3.3	
$\dot{L}_{eff}(t) = \dfrac{[L(t) - L_{eff}(t)]^3}{T_L(1 - Af)}$	T_L(ms) 0.088				T_L(ms) (same as slow twitch)			
$\dot{S}(t, f_{eff}) = \dfrac{a_s - S(t)}{T_s}, \quad a_s = \begin{cases}a_{S1}, f_{eff}(t) < 0.1 \\ a_{S2}, f_{eff}(t) \geq 0.1\end{cases}$	a_{s1} 1.0	a_{s2} 1.0	T_S(ms) –		a_{s1} 1.76	a_{s2} 0.96	T_S(ms) 43	
$\dot{Y}(t) = \dfrac{1 - c_Y[1 - \exp(-abs\|V\|/V_Y)] - Y(t)}{T_Y}$	c_Y 0.35	V_Y 0.1	T_Y(ms) 200		c_Y 0	V_Y –	T_Y(ms) –	
$\dot{f}_{int}(t, f_{env}, L) = \dfrac{f_{env}(t) - f_{int}(t)}{T_{\hat{f}}}, \quad \dot{f}_{eff}(t, f_{int}, L) = \dfrac{f_{int}(t) - f_{eff}(t)}{T_f}, \quad T_f = \begin{cases}T_{f1}L^2 + T_{f2}f_{env}(t), & \dot{f}_{eff}(t) \geq 0 \\ (T_{f3} + T_{f4}Af)/(L), & \dot{f}_{eff}(t) < 0\end{cases}$	T_{f1}(ms) 24.2	T_{f2}(ms) 16	T_{f3}(ms) 33.2	T_{f4}(ms) 17.8	T_{f1}(ms) 20.6	T_{f2}(ms) 13.6	T_{f3}(ms) 28.2	
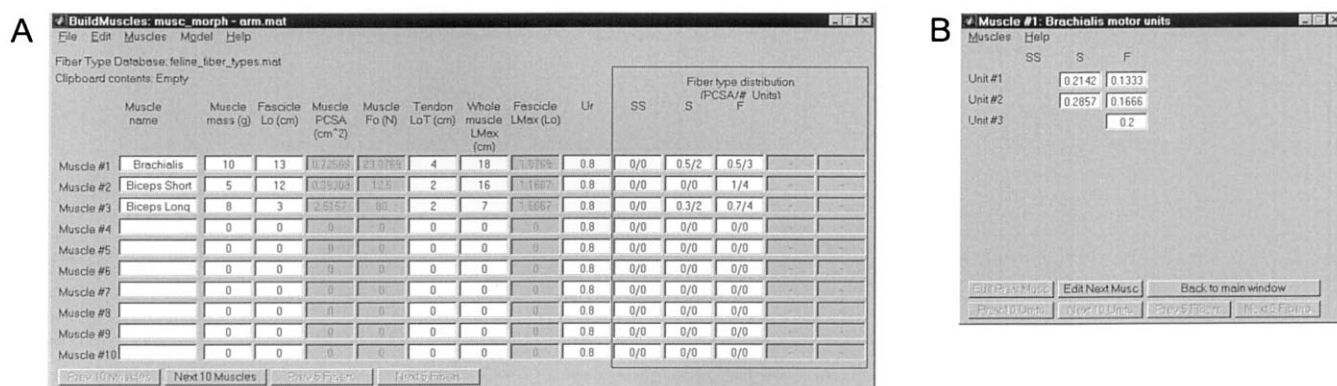
Fig. 5. (A) 'Screen shot' of the BuildMuscles function. Parameters for each muscle are modified in this main window. Each muscle is entered into a separate row. The left-most columns accept muscle morphometry measures for each muscle, while the right-most columns (enclosed by black frame) describe its composition by fiber-type and numbers of modeled units. (B) Secondary window of BuildMuscles function for displaying and editing fractional PCSA of each motor unit. Motor units of a single fiber type are arranged in columns and recruited sequentially from the top down.

4.3. Musculotendon blocks

The BuildMuscles function ultimately is used to create Simulink blocks for each muscle you have chosen to model, based on the parameters currently entered in the BuildMuscles function and the selected fiber type database file. These Simulink blocks are stand-alone and, once created, can run on any supported version of Simulink, even without the Virtual Muscle package installed. A potential drawback to making the Simulink blocks stand-alone is that changes made to the database files will not be reflected automatically in the existing blocks. To address this issue, a feature has been included in the BuildMuscles function that will allow users easily to replace muscle blocks in existing Simulink model files with updated muscle blocks of the same name. This facilitates building a complex simulation and then examining the consequences of making various changes to the muscle properties.

As mentioned previously, the Simulink musculotendon blocks require inputs for activation, typically between 0 and 1, and for musculotendon path length, in centimeters. In Simulink, these inputs can easily be driven by constants, ramp or sinusoidal inputs, feedback or feedforward mechanisms, external data files, or even outputs of other software packages. The latter driving mechanism is explained in a following section. The output from the musculotendon element is force in Newtons. This force output can be used as inputs to other Simulink blocks, directed to output files or display scopes, or directed to external applications.

5. Application

5.1. Scaling EMG to neural activation

Recorded EMG envelopes are commonly used to drive contractile elements in musculoskeletal models, with the assumption that EMG reflects the underlying muscle activation. Frequently, EMG amplitudes are scaled linearly (e.g. by maximal voluntary contraction, Perry and Bekey, 1981) to obtain activation. This is reasonable in theory; an action potential in a motor unit should produce an electrical signal whose amplitude corresponds roughly to the PCSA of its muscle fibers, assuming the recorded EMG signal captures the sum of all synchronous currents in each muscle fiber. In reality, the precise relationship between EMG and activation is complex and specific to the recording equipment and implantation techniques used. Occlusion (the tendency of temporally overlapping action potentials from different motor units to cancel one another depending on the phase of their signal) complicates this assumption of linearity. Occlusion increases as more units become simultaneously active; thus recorded EMG would increase at a less-than-linear rate relative to activation. There is at least one piece of experimental evidence that suggests that a linear relationship is reasonable, however. A linearly scaled EMG envelope recorded using implanted bipolar electrodes was injected intracellularly into motoneurons and succeeded in reproducing the frequency modulation of single motor units recorded along with the EMG envelope (Hoffer et al., 1987a).

For the present model, we recommend adjusting the scaling between EMG and activation to suit the type of model being simulated. For a Virtual Muscle block consisting of a single unit, there will be a roughly linear increase in force with activation. On the other hand, a multiple-motor unit model will result in a greater-than-linear increase in force with activation; this is an emergent property of the recruitment model utilized. This occurs because an increasing activation will increase linearly the firing frequency envelopes of any currently recruited motor units, while simultaneously recruiting

additional motor units at their respective recruitment thresholds. The effective muscle activation versus input activation is illustrated in Fig. 6. Effective muscle activation corresponds to the total firing frequency of all motor units weighted by the size of each motor unit, which in turn corresponds roughly to the amount of force produced by the muscle in a steady-state, isometric contraction. In the case of multiple motor unit simulations, taking the square root of a signal before using it as an activation input will restore an approximately linear relationship with total effective muscle activation. As noted above, this effect may occur automatically in EMG signals as a result of occlusion.

5.2. Guidelines for defining musculotendon elements

Each musculotendon element acts independently and singularly; i.e. it generates a single level of force in response to its own activation and length inputs. It may be beneficial, however, to use more or less than one such modeled element to represent each given anatomical muscle. For example, researchers often model movements at two joints using a total of six muscle elements (e.g. Karniel and Inbar, 1997; Gomi and Osu, 1999), an elbow flexor and extensor, a shoulder flexor and extensor, and a biarticular flexor and extensor. In this case, the level of accuracy in representing the arm is reduced for the purpose of simplification. On the other hand, when greater detail is required, it may be necessary to represent a single anatomical muscle with more than one simulated musculotendon element. For example, it has been shown that different subpopulations of motor units in one region of feline sartorius muscle are recruited under different conditions; one subpopulation participates only during stance phase and another only

during the swing phase (Hoffer et al., 1987b). Because these so-called 'task groups' may have different sources of control signals, each region may be modeled with a separate musculotendon element. A similar situation may arise in mechanically heterogeneous muscle, for example in the feline biceps femoris (Chanaud et al., 1991). In this muscle, the anterior region has a moment about the hip joint, while the posterior region mechanically favors action as a knee flexor; it was found that these regions had different kinematics and were differentially recruited depending on the behavior performed.

The required morphometric measures for the creation of each musculotendon element are often available in the literature. Measurements for many muscles in humans are summarized by Yamaguchi et al. (1990), and additional data on the human upper limb have been published by Veeger et al. (1997). Detailed histochemical fiber distributions for many individual muscles can also be found (e.g. Ariano, 1973; Johnson et al., 1973; Roy et al., 1984).

5.3. Interfacing with external applications

Researchers who are already running simulations in Matlab and Simulink likely already have a method for modeling the dynamics of the relevant portion of the skeletal system. However, for those users who have not yet created a simulation and do not wish to write their own linked-segment dynamics model in Simulink, and for those who use a commercial kinetic modeling software, we have provided an example of how to interface Matlab with external applications. The sample provided is for use with Working Model 2D (MSC.Working Knowledge, San Mateo, CA), a dynamics modeling package for personal computers run-
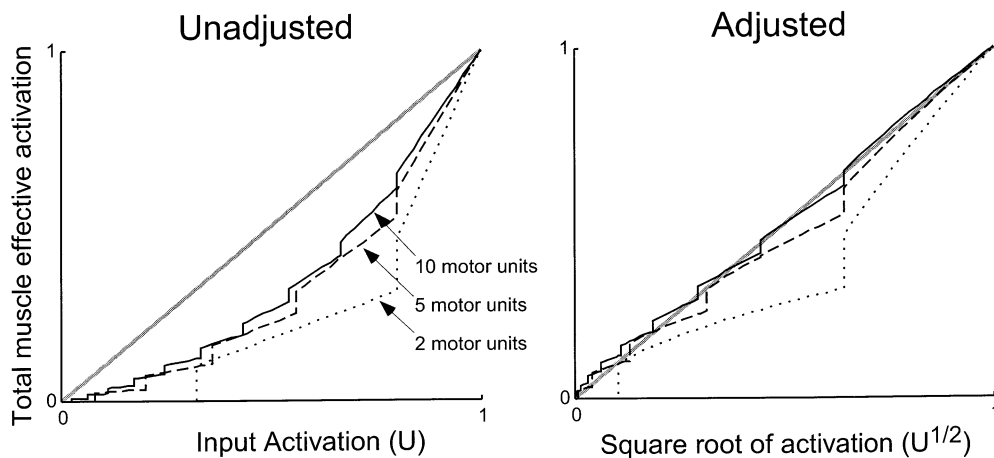

Fig. 6. Emergent properties of the recruitment scheme when used with two, five and ten motor units (dotted, dashed and solid black lines, respectively). Effective activation of all motor units versus the activation input signal is plotted. Effective activation represents the total firing frequency of all motor units in muscle weighted by PCSA of each motor unit. At left, with an unadjusted, linearly increasing activation, the effective activation rises slowly at first and more rapidly later. This deviates strongly from a linear response (thick gray line). At right, when the activation signal is adjusted to its square root, effective activation approaches linear.
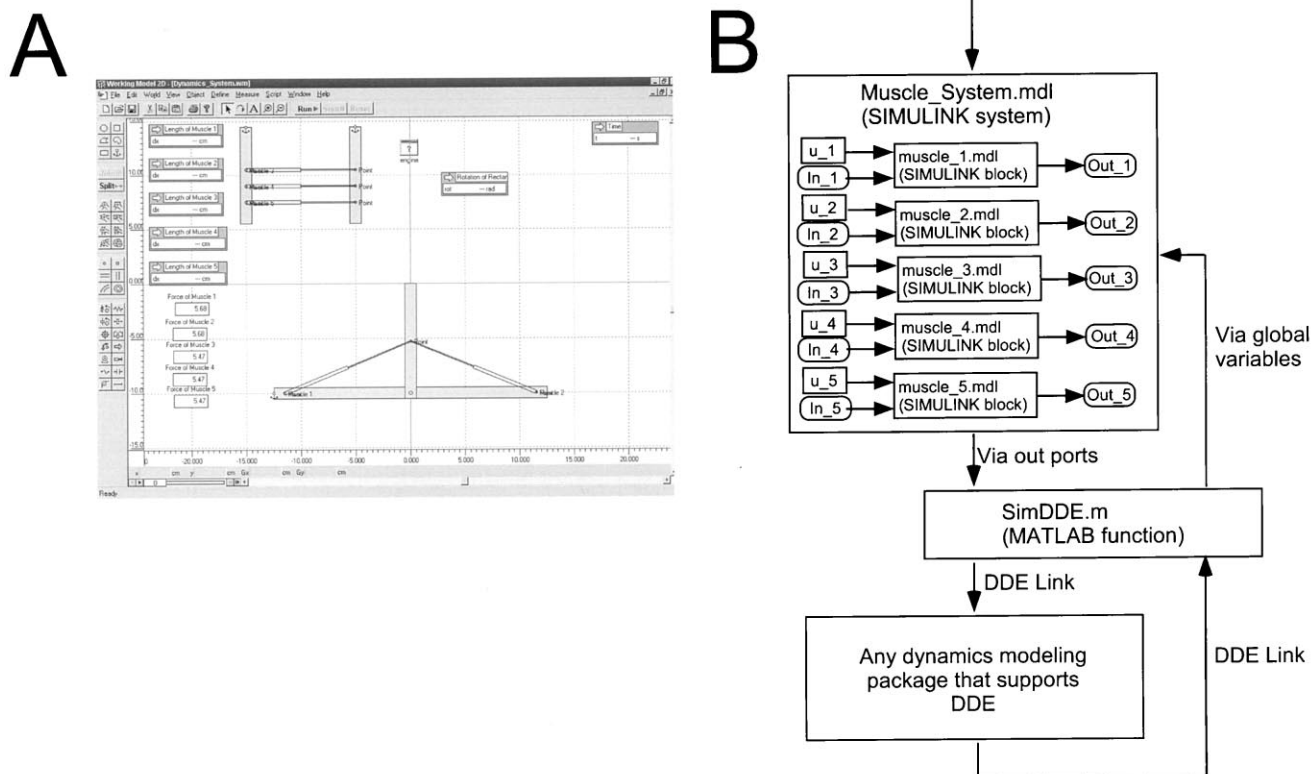
Fig. 7. (A) 'Screen shot' of Working Model 2D running the included sample Dynamics System. Five musculotendon actuators are depicted; two are attached to the simplified joint model, while the remaining three are not in use. Also displayed are various meters displaying information on actuator lengths and the force that each actuator element exerts. (B) Schematic representation of the structure of the interface between the Virtual Muscle elements (at top) and any external dynamics modeling package (at bottom). The SimDDE function acts as the interface between the two packages.

ning under Microsoft Windows (Fig. 7). Our inter-application link utilizes DDE, an interface supported by many Windows-based applications.

The sample provided uses a wrapper function that runs in Matlab called SimDDE, which functions to pass data from five Virtual Muscle musculotendon elements in Simulink to Working Model 2D, and vice-versa (Fig. 7). The SimDDE function is called once per simulation time-step by Working Model 2D, which in turn simulates activation (specified in Simulink) and length changes (passed from Working Model 2D) in the Virtual Muscle blocks, and returns the forces produced by each musculotendon element. The SimDDE application is fully functional, and any user who wishes to use up to five musculotendon elements can immediately start using the Virtual Muscle package without further programming. The muscle database associated with the SimDDE model can be assigned the appropriate muscle morphometry, and the actuator elements within Working Model 2D can be dragged onto the desired skeletal segments. The simulation is then ready to run from within Working Model 2D.

6. Discussion

Virtual Muscle should integrate well with existing simulations designed in Matlab or Simulink, especially if the simulations have been designed hierarchically, with separate elements to represent muscle properties. In this case, it may be as simple as replacing an existing representation of musculotendons with Simulink blocks generated by Virtual Muscle. On the other hand, simulations that lack explicit, independent muscle elements will not benefit from the Virtual Muscle package as easily. These simulations are often driven by optimization criteria relating to net joint torque or stiffness values or cost functions but do not specify explicitly the constituent muscles. In these cases, models would have to be significantly redesigned to use realistic musculotendon elements. Software and documentation are available free via the Internet at http://ami.usc.edu/mddf/virtualmuscle.

Before we created this system to allow the rapid modification of muscle properties in Matlab and Simulink, we had implemented a similar muscle model in Working Model 2D alone to examine the implications

of realistic muscle properties for motor control (Loeb et al., 1999). Because Working Model 2D is essentially a segment dynamics package, entering the many mathematical equations for each muscle was labor-intensive, and modifications had to be made manually for each individual muscle. Modeling multiple motor units with type-specific properties and independent frequency modulation was impractical. Since then, we have used the Virtual Muscle package described here to create and examine a biomechanical model of the arm entirely in Matlab and Simulink, using eight anatomically accurate muscles (Cheng and Loeb, in preparation).

Interfacing with an external dynamics application such as Working Model 2D obviates the need to code explicitly a model of segment dynamics. It is unavoidable, however, that running these two applications at the same time is less computationally efficient than having the muscle and segment dynamics model running in a single package. Thus, users wishing to begin using a motor control simulation need to decide on a balance between performance and simplicity. Similarly, decisions need to be made when creating the musculotendon elements as to the level of complexity desired. Each additional motor unit simulated will increase the computational time required, and users need to trade-off speed versus accuracy. Future directions include adapting portions of each Virtual Muscle block to run as compiled Matlab code to decrease computational time. While this would improve efficiency, it would be at the expense of cross-platform compatibility.

Acknowledgements

Appendix A. Refining the original fiber type model

Our previous work in the two feline muscles soleus and caudofemoralis revealed that the passive force length relationship (F_{PE1}) did not scale with L_0 between muscle specimens, motivating us to publish several examples of parameters for different specimens (Brown et al., 1996b, 1999). Separate work on these and other hindlimb muscles of the cat revealed that F_{PE1} scales more appropriately with the maximal anatomical length of a muscle, L_{max} (Brown et al., 1996a). We therefore, combined passive force length data from a total of 61 feline hindlimb muscles to determine a single best-fit relationship based upon data that was normalized to

L_{max} (data not reproduced here; soleus data from Scott et al., 1996, semitendinous, sartorius, tenuissimus, biceps femoris anterior and caudofemoralis data from Brown et al., 1996a, additional caudofemoralis data from Brown et al., 1999). Because of the steep slope of this relationship and the variability in the x-direction, curve-fitting was accomplished in two steps. First, all data were shifted in the x-direction (i.e. length) so that they aligned at 0.5 F_0. The constants responsible for the curvature (c_1 and k_1; see Table 1) were then fit to this shifted data using the standard Levenberg–Marquardt algorithm. Using these values of c_1 and k_1, the function was then fit to the original non-shifted data to estimate the best-fit value for L_{r1}, minimizing error in the x-direction (i.e. length). We have also added a small viscosity for stability purposes during simulation.

$$F_{PE1}(L) = c_1 k_1 \ln\left\{\exp\left[\frac{(L/L_{max} - L_{r1})}{k_1}\right] + 1\right\} + \eta V \quad (1)$$

Equation adapted from Brown et al. (1999). L and L_{max} must be in the same units (usually L_0). Force output is in units of F_0.

We have also re-fit our fast-twitch FV relationship. The original data from caudofemoralis (Fig. 6 in Brown et al., 1999) was limited to slow to moderate speeds. Because our curve fitting was not constrained by a realistic V_{max}, the curve extrapolated to a V_{max} of only 7.4 L_0/s. Spector et al. (1980) measured V_{max} values for feline soleus and medial gastrocnemius of 4.8 and 12.8 L_0/s, respectively. We therefore, re-fit our feline caudofemoralis FV parameters but with V_{max} fixed at 14 L_0/s (14 L_0/s was chosen because caudofemoralis is 100% fast-twitch and thus is likely a little faster than medial gastrocnemius). This change had a negligible effect on the model's force estimation when compared with the original curve derived from the available data for slow to moderate speeds, but it now extrapolates to a more plausible estimate for fast-twitch V_{max}. The new parameters for human muscle are listed in Table 1. (To calculate the new feline FV fast-twitch parameters, simply scale the two parameters V_{max} and b_V by the equivalent change in $V_{0.5}$ — or let Virtual Muscle do it automatically).

Appendix B. Estimation of muscle model parameters for human skeletal muscle fiber types

Because of the difficulty in performing experiments on human muscles, we have chosen to extrapolate a detailed model of force production in feline muscle described previously (Brown and Loeb, 2000; Brown et al., 1999) in order to create a similar model for human skeletal muscle. The extrapolated parameters for human skeletal muscle fiber types are listed along with the associated equations in Table 1. We extrapolated the

model to human fiber types using the tools described in this paper, namely by changing optimal sarcomere length (which scales FL and F_{PE2}), $V_{0.5}$ (which scales FV) and $f_{0.5}$ (which scales f_{eff}, which controls rise and fall times). This appendix describes how we estimated these values for human fiber types. Although 'sag' is not evident in some human fiber types, (thenar units, Thomas et al., 1991; toe extensor units, Macefield et al., 1996) it is clearly evident in others (triceps surae; van Zandwijk et al., 1998) so we have included it until further evidence can clarify this issue. The limited data for 'yielding' suggest that it, too, is present in at least some human muscles (erector spinae, Sutarno and McGill, 1995).
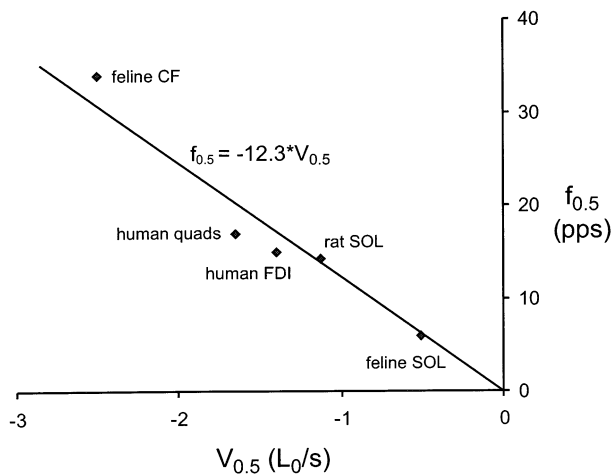

Fig. 8. Frequency–velocity relationship. Relative measures of the frequency ($f_{0.5}$) and velocity ($V_{0.5}$) are plotted against each other for five muscles, two each from cat and human and one from rat. ($f_{0.5}$ is defined as the stimulus frequency necessary to produce 0.5 F_0 at 1.0 L_0 during isometric conditions, while $V_{0.5}$ is the shortening velocity necessary to produced 0.5 F_0 at 1.0 L_0 during maximal (tetanic stimulation). The two feline muscles are the exclusively fast-twitch caudofemoralis (CF) and the exclusively slow-twitch soleus (SOL). The rat muscle is the mostly show SOL. The two human muscles are of mixed composition; first dorsal interossus (FDI) and the quadriceps (quads). The data were garnered from the following studies; feline CF $V_{0.5}$ and $f_{0.5}$ from Brown et al. (1999); feline SOL $V_{0.5}$ from Brown et al. (1996a,b) and $f_{0.5}$ from Rack and Westbury (1969) as described in Brown et al. (1999); rat SOL $V_{0.5}$ from Ranatunga (1984) and $f_{0.5}$ from Binder-Macleod and Barrish (1992) — note, original data were collected at the optimal twitch length which is approximately 1.1 L_0 (Roszek et al., 1994; Brown and Loeb, 1998) and so we shifted the reported force–frequency relationship assuming the same length dependence as for feline SOL (Brown et al., 1999); human quads $V_{0.5}$ from Marshall et al. (1990) converted from cm to L_0 using an L_0 estimate of 9 cm from Scott et al. (1993) and $f_{0.5}$ from Edwards et al. (1977) — note, original data collected at a knee angle of 90, which is approximately 1.1 L_0 based on data from Marshall et al. (1990), therefore, we shifted the reported force-frequency relationship as per rat SOL above; human FDI $V_{0.5}$ from Cook and McDonagh (1996) and $f_{0.5}$ averaged from Milner-Brown et al. (1973a,b), Rutherford and Jones (1988).

Optimal sarcomere lengths for various species have been published previously by Herzog et al. (1992). We used their measured values of 2.4 and 2.7 μm for cat and human, respectively, to scale the active and passive force–length properties.

Estimates of $V_{0.5}$ and $f_{0.5}$ were more difficult to obtain. Previous work has shown that V_{max} is approximately proportional to contraction time for fiber types from a wide range of species (Close, 1972). Other work has shown that contraction time is proportional to $f_{0.5}$. (Kernell et al., 1983; Botterman et al., 1986). Together these imply that $V_{0.5}$ may be proportional to $f_{0.5}$ across species. In Fig. 8 we plot $V_{0.5}$ versus $f_{0.5}$ for two feline muscles composed exclusively of fast-twitch and slow-twitch muscle (upon which the our feline muscle model was based), one rat muscle of mostly slow-twitch fiber type composition and two human muscles of mixed fiber type composition (see legend for details). Given the similarity in ratios between the various muscles across fiber types and across species, we have assumed that for normal muscles the $f_{0.5}/V_{0.5}$ ratio is constant at a value of 12(pulses/L_0).

Based upon this notion of a constant ratio, we need only acquire $f_{0.5}$ or $V_{0.5}$ to define the properties of a 'typical' fiber type. Single unit data for human muscles are currently extremely sparse and thus unlikely to represent quantitatively the populations as a whole. Therefore, we have used whole-muscle values along with fiber type estimations from Johnson et al. (1973) to estimate different fiber type speeds as a first approximation. For adductor pollicus ($\sim$ 80% type I), $f_{0.5}$ is $\sim$ 14–18 pps (Bigland and Lippold, 1954; Cooper et al., 1988; Rutherford and Jones, 1988). For the quadriceps ($\sim$ 45% type I) $f_{0.5}$ is 16–18 pps (Edwards et al., 1977 — note; original data were collected at a knee angle of 90°, which is approximately 1.1 L_0 (Marshall et al., 1990); therefore, we shifted the reported force–frequency relationship assuming the same length dependence as feline muscles (Brown et al., 1999)). For first dorsal interosseus ($\sim$ 55% type I) $f_{0.5}$ is 12–18 pps, averaged from Milner-Brown et al. (1973b), Rutherford and Jones (1988). For the triceps surae ($\sim$ 60% type I) $f_{0.5}$ is $\sim$ 10–12 pps (van Zandwijk et al., 1998). Based upon these limited data, we tentatively suggest three fiber types, a 'super' slow-twitch with $f_{0.5}$ of $\sim$ 6 pps, a 'typical' slow-twitch with $f_{0.5}$ of $\sim$ 12 pps, and a fast-twitch with $f_{0.5}$ of 20 pps (see Fig. 3B). $V_{0.5}$ was estimated at 1/12th those values as described above. We assume that only the triceps surae muscles contain the 'super slow' fiber type (analogous to feline soleus) and that the other muscles described here contain the 'typical' slow fiber type. When better data become available, we should be able to improve upon these crude estimates.

References

Ariano MA, Armstrong RB, Edgerton VR. Hindlimb muscle fiber populations of five mammals. J Histochem Cytochem 1973;21:51–5.

Asatryan DG, Feldman AG. Functional tuning of the nervous system with control of movement or maintenance of a steady posture. I. Mechanographic analysis of the work of the joint or execution of a postural task. Biofizika 1965;10:837–46.

Balnave CD, Allen DG. The effect of muscle length on intracellular calcium and force in single fibers from mouse skeletal muscle. J Physiol 1996;492:705–13.

Bigland B, Lippold OCJ. Motor unit activity in the voluntary contraction of human muscle. J Physiol 1954;125:322–35.

Binder-Macleod SA, Barrish WJ. Force response of rat soleus muscle to variable-frequency train stimulation. J Neurophysiol 1992;68:1068–78.

Botterman BR, Iwamoto GA, Gonyea WJ. Gradation of isometric tension by different activation rates in motor units of cat flexor carpi radialis muscle. J Neurophysiol 1986;56:494–506.

Brown IE. Measured and modeled properties of mammalian skeletal muscle. Ph.D. thesis, Queen's University, 1998.

Brown IE, Loeb GE. Post-activation potentiation — a clue for simplifying models of muscle dynamics. Am Zool 1998;38:743–54.

Brown IE, Loeb GE. Measured and modeled properties of mammalian skeletal muscle: IV. Dynamics of activation and deactivation, J Muscle Res Cell Motil 2000;21:33–47.

Brown IE, Liinamaa TL, Loeb GE. Relationships between range of motion, L_0 and passive force in five strap-like muscles of the feline hindlimb. J Morphol 1996a;230:69–77.

Brown IE, Scott SH, Loeb GE. Mechanics of feline soleus: II. Design and validation of a mathematical model. J Muscle Res Cell Motil 1996b;17:219–32.

Brown IE, Satoda T, Richmond FJR, Loeb GE. Feline caudofemoralis muscle. Muscle fiber properties, architecture, and motor innervation. Exp Brain Res 1998;121:76–91.

Brown IE, Cheng EJ, Loeb GE. Measured and modeled properties of mammalian skeletal muscle: II. The effects of stimulus frequency on force–length and force–velocity relationships. J Muscle Res Cell Motil 1999;20:627–43.

Chanaud CM, Pratt CA, Loeb GE. Functionally complex muscles of the cat hindlimb. V. The roles of histochemical fiber-type regionalization and mechanical heterogeneity in differential muscle activation. Exp Brain Res 1991;85:300–13.

Close R. Dynamic properties of mammalian skeletal muscle. Physiol Rev 1972;52:129–79.

Cook CS, McDonagh MJ. Force responses to constant-velocity shortening of electrically stimulated human muscle–tendon complex. J Appl Physiol 1996;81:384–92.

Cooper RG, Edwards RHT, Gibson H, Stokes MJ. Human muscle fatigue: frequency dependence of excitation and force generation. J Physiol 1988;397:585–99.

De Luca CJ, Foley PJ, Erim Z. Motor unit control properties in constant-force isometric contractions. J Neurophysiol 1996;76:1503–16.

Durfee WK, Palmer KI. Estimation of force–activation, force–length, and force–velocity properties in isolated, electrically stimulated muscle. IEEE Trans Biomed Eng 1994;41:205–16.

Edwards RH, Young A, Hosking GP, Jones DA. Human skeletal muscle function: description of tests and normal values. Clin Sci Mol Med 1977;52:283–90.

Georgopoulos AP. On reaching. Annu Rev Neurosci 1986;9:147–70.

Gomi H, Osu R. Task-dependent viscoelasticity of human multijoint arm and its spatial characteristics for interaction with environments. J Neurosci 1999;18:8965–78.

Gonzalez RV, Hutchins EL, Barr RE, Abraham LD. Development and evaluation of a musculoskeletal model of the elbow joint complex. J Biomech Eng 1997;118:32–40.

Gottlieb GL. On the voluntary movement of compliant (inertial-viscoelastic) loads by parcellated control mechanisms. J Neurophysiol 1996;76:3207–29.

Henneman E. Organization of the spinal cord. In: Mountcastle B, editor. Medical Physiology, 12th ed. St. Louis: C.V. Mosby Co., 1968:1717–32.

Herzog W, Kamal S, Clarke HD. Myofilament lengths of cat skeletal muscle: theoretical considerations and functional implications. J Biomech 1992;25:945–8.

Hoffer JA, Sugano N, Loeb GE, Marks WB, O'Donovan MJ, Pratt CA. Cat hindlimb motoneurons during locomotion: II. Normal activity patterns. J Neurophysiol 1987a;57:530–53.

Hoffer JA, Loeb GE, Sugano N, Marks WB, O'Donovan MJ, Pratt CA. Cat hindlimb motoneurons during locomotion. III. Functional segregation in sartorius. J Neurophysiol 1987b;57:554–62.

Hogan N. The mechanics of multi-joint posture and movement control. Biol Cybern 1985;52:315–31.

Johnson MA, Polgar J, Weightman D, Appleton D. Data on the distribution of fibre types in thirty-six human muscles. An autopsy study. J Neurol Sci 1973;18:111–29.

Karniel A, Inbar GF. A model for learning human reaching movements. Biol Cybern 1997;77:173–83.

Kaufman KR, An K-N, Litchy WJ, Chao EYS. Physiological prediction for muscle forces I. Application to isokinetic exercise. Neuroscience 1991;40:793–804.

Kernell D, Eerbeek O, Verhey BA. Relation between isometric force and stimulus rate in cat's hindlimb motor units of different twitch contraction times. Exp Brain Res 1983;50:220–7.

Loeb GE, Brown IE, Cheng EJ. A hierarchical foundation for models of sensorimotor control. Exp Brain Res 1999;126:1–18.

Lucas SM, Ruff RL, Binder MD. Specific tension measurements in single soleus and medial gastrocnemius muscle fibers of the cat. Exp Neurol 1987;95:142–54.

Macefield VG, Fuglevand AJ, Bigland-Ritchie B. Contractile properties of single motor units in human toe extensors assessed by intraneural motor axon stimulation. J Neurophysiol 1996;75:2509–19.

Marshall RN, Mazur SM, Taylor NA. Three-dimensional surfaces for human muscle kinetics. Eur J Appl Physiol 1990;61:263–70.

Milner-Brown HS, Stein RB, Yemm R. The orderly recruitment of human motor units during voluntary isometric contractions. J Physiol 1973a;230:359–70.

Milner-Brown HS, Stein RB, Yemm R. Changes in firing rate of human motor units during linearly changing voluntary contractions. J Physiol 1973b;230:371–90.

Osu R, Gomi H. Multijoint muscle regulation mechanisms examined by measured human arm stiffness and EMG signals. J Neurophysiol 1999;81:1458–68.

Otten E. A myocybernetic model of the jaw system of the rat. J Neurosci Methods 1987;21:287–302.

Perry J, Bekey GA. EMG–force relationship in skeletal muscle. Crit Rev Biomed Eng 1981;12:1–22.

Rack PMH, Westbury DR. The effects of length and stimulus rate on tension in the isometric cat soleus muscle. J Physiol 1969;204:443–60.

Ranatunga KW. The force–velocity relation of rat fast- and slow-twitch muscles examined at different temperatures. J Physiol 1984;351:517–29.

Roszek B, Baan GC, Huijing PA. Decreasing stimulation frequency-dependent length–force characteristics of rat muscle. J Appl Physiol 1994;77:2115–24.

Roy RR, Bello MA, Powell PL, Simpson DR. Architectural design and fiber-type distribution of the major elbow flexors and extensors of the monkey (*Cynomolgus*). Am J Anat 1984;171:285–93.

Rutherford OM, Jones DA. Contractile properties and fatiguability of the human adductor pollicis and first dorsal interosseus: a comparison of the effects of two chronic stimulation patterns. J Neurol Sci 1988;85:319–31.

Scott SH, Loeb GE. Mechanical properties of the aponeurosis and tendon of the cat soleus muscle during whole-muscle isometric contractions. J Morphol 1995;224:73–86.

Scott SH, Engstrom CM, Loeb GE. Morphometry of human thigh muscles. Determination of fascicle architecture by magnetic resonance imaging. J Anat 1993;182:249–57.

Scott SH, Brown IE, Loeb GE. Mechanics of feline soleus: I. Effect of fascicle length and velocity on force output. J Muscle Res Cell Motil 1996;17:207–19.

Shue G, Crago PE, Chizeck HJ. Muscle-joint models incorporating activation dynamics, moment–angle, and moment–velocity properties. IEEE Trans Biomed Eng 1995;42:212–22.

Soechting JF, Flanders M. Evaluating an integrated musculoskeletal model of the human arm. J Biomech Eng 1997;119:93–102.

Spector SA, Gardiner PF, Zernicke RF, Roy RR, Edgerton VR. Muscle architecture and force–velocity characteristics of cat soleus and medial gastrocnemius: implications for motor control. J Neurophysiol 1980;44:951–60.

Sutarno CG, McGill SM. Isovelocity investigation of the lengthening behavior of the erector spinae muscles. Eur J Appl Physiol 1995;70:146–53.

Thomas CK, Johansson RS, Bigland-Ritchie BR. Attempts to physiologically classify human thenar motor units. J Neurophysiol 1991;65:1501–8.

Todorov E. Direct cortical control of muscle activation in voluntary arm movements: a model. Nature Neurosci 2000;3:391–8.

van der Helm FCT. Analysis of the kinematic and dynamic behavior of the shoulder mechanism. J Biomech 1994;27:527–50.

van Zandwijk JP, Bobbert MF, Harlaar J, Hof AL. From twitch to tetanus for human muscle: experimental data and model predictions for m-triceps surae. Biol Cybern 1998;79:121–30.

Veeger HEJ, Yu B, An K, Rozendal RH. Parameters for modeling the upper extremity. J Biomech 1997;30:647–52.

Yamaguchi GT, Sawa AGU, Moran DW, Fessler MJ, Winters JM. Appendix: a survey of human musculotendon actuator parameters. In: Winters JM, Woo SL-Y, editors. Multiple Muscle Systems: Biomechanics and Movement Organization. New York: Springer, 1990:717–73.

Zajac FE. Muscle and tendon, properties, models, scaling and application to biomechanics and motor control. Crit Rev Biomed Eng 1989;17:359–411.

Update

Erratum

Erratum to 'Virtual muscle: a computational approach to understanding the effects of muscle properties on motor control'☆ [J. Neurosci. Methods 101 (2000) 117–130]

Ernest J. Cheng [1], Ian E. Brown, Gerald E. Loeb *

Department of Biomedical Engineering, DRB-B12, Medical Device Development Facility, Alfred E. Mann Institute for Biomedical Engineering, University of Southern California, MC 1112, 1042 West 36th Place, Los Angeles, CA 90089-1112, USA

The publisher regrets that a number of typographical errors occurred in Table 1 of the above mentioned manuscript. Please find the corrected table overleaf.

Table 1
Summary of model equations and best-fit constants

Curve	Typical slow twitch fibers	Fast-twitch fibers
$F_{SE}(L^T) = c^T k^T \ln\left\{\exp\left[\dfrac{(L^T - L_r^T)}{k^T}\right] + 1\right\}$	c^T 27.8 ; k^T 0.0047 ; L_r^T 0.964	c^T (same as slow twitch) ; k^T (same as slow twitch) ; L_r^T
$F_{PE1}(L) = c_1 k_1 \ln\left\{\exp\left[\dfrac{(L/L_{max} - L_{r1})}{k_1}\right] + 1\right\} + \eta V$	c_1 23.0 ; k_1 0.046 ; L_{r1} 1.17 ; η 0.01	c_1 (same as slow twitch) ; L_{r2}
$F_{PE2}(L) = c_2\{\exp[k_2(L - L_{r2})] - 1\},\quad F_{PE2} \leq 0$	c_2 −0.020 ; k_2 −21.0 ; L_{r2} 0.70	c_2 (same as slow twitch) ; k_2 (same as slow twitch) ; L_{r2}
$FL(L) = \exp\left(-\text{abs}\left|\dfrac{L^B - 1}{\omega}\right|^{\rho}\right)$	ω 1.12 ; β 2.30 ; ρ 1.62	ω 0.75 ; β 1.55 ; ρ 2.12
$FV(V, L) = \begin{cases} (V_{max} - V)/(V_{max} + (c_{v0} + c_{v1}L)V), & V \leq 0 \\ (b_v - (a_{v0} + a_{v1}L + a_{v2}L^2)V)/(b_v + V), & V > 0 \end{cases}$	V_{max} −7.88 ; c_{v0} 5.88 ; c_{v1} 0 ; b_v 0.35 ; a_{v0} −4.70 ; a_{v1} 8.41 ; a_{v2} −5.34	V_{max} −9.15 ; c_{v0} −5.70 ; c_{v1} 9.18 ; b_v 0.69 ; a_{v0} −1.53 ; a_{v1} 0 ; a_{v2} 0
$Af(f_{eff}, L_{eff}, Y, S) = 1 - \exp\left[-\left(\dfrac{YSf_{eff}}{a_f n_f}\right)^{n_f}\right],\quad n_f = n_{f0} + n_{f1}\left(\dfrac{1}{L_{eff}} - 1\right)$	a_f 0.56 ; n_{f0} 2.1 ; n_{f1} 5	a_f 0.56 ; n_{f0} 2.1 ; n_{f1} 3.3
$\dot{L}_{eff}(t) = \dfrac{[L(t) - L_{eff}(t)]^3}{T_L(1 - Af)}$	T_L(ms) 0.088	T_L(ms) (same as slow twitch)
$\dot{S}(t, f_{eff}) = \dfrac{a_s - S(t)}{T_s},\quad a_s = \begin{cases} a_{s1}, f_{eff}(t) < 0.1 \\ a_{s2}, f_{eff}(t) \geq 0.1 \end{cases}$	a_{s1} 1.0 ; a_{s2} 1.0 ; T_S(ms) –	a_{s1} 1.76 ; a_{s2} 0.96 ; T_S(ms) 43
$\dot{Y}(t) = \dfrac{1 - c_Y[1 - \exp(-\text{abs}|V|/V_Y)] - Y(t)}{T_Y}$	c_Y 0.35 ; V_Y 0.1 ; T_Y(ms) 200	c_Y 0 ; V_Y – ; T_Y(ms) –
$\dot{f}_{int}(t, f_{env}, L) = \dfrac{f_{env}(t) - f_{int}(t)}{T_f},\quad T_f = \begin{cases} T_{f1}L^2 + T_{f2}f_{env}(t), & \dot{f}_{eff}(t) \geq 0 \\ (T_{f3} + T_{f4}Af)/(L), & \dot{f}_{eff}(t) < 0 \end{cases}$ $\dot{f}_{eff}(t, f_{int}, L) = \dfrac{f_{int}(t) - f_{eff}(t)}{T_f}$	T_{f1}(ms) 34.3 ; T_{f2}(ms) 22.7 ; T_{f3}(ms) 47.0 ; T_{f4}(ms) 25.2	T_{f1}(ms) 20.6 ; T_{f2}(ms) 13.6 ; T_{f3}(ms) 28.2 ; T_{f4}(ms) 15.1