



# Muscle mass in musculoskeletal models

Dinesh K. Pai

Sensorimotor Systems Laboratory, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada

## ARTICLE INFO

Article history:  
Accepted 7 April 2010

Keywords:  
Dynamics  
Simulation  
Musculoskeletal  
Muscle  
Inertia  
Mass

## ABSTRACT

Most current models of musculoskeletal dynamics lump a muscle's mass with its body segment, and then simulate the dynamics of these body segments connected by joints. As shown here, this popular approach leads to errors in the system's inertia matrix and hence in all aspects of the dynamics. Two simplified mathematical models were created to capture the relevant features of monoarticular and biarticular muscles, and the errors were analyzed. The models were also applied to two physiological examples: the triceps surae muscles that plantar flex the human ankle and the biceps femoris posterior muscle of the rat hind limb. The analysis of errors due to lumping showed that these errors can be large. Although the errors can be reduced in some postures, they cannot be easily eliminated in models that use segment lumping. Some options for addressing these errors are discussed.

© 2010 Elsevier Ltd. All rights reserved.

## 1. Introduction

Computer simulation of the dynamics of musculoskeletal systems is an important tool in biomechanics. It is useful for a wide range of problems, such as predicting the outcome of tendon transfers (Delp and Loan, 2000), understanding multijoint movements (Hollerbach and Flash, 1982), evaluating ergonomics (AnyBodyTech, 2009), optimizing biomechanical performance (Pandy et al., 1990; Kargo et al., 2002), developing neural prostheses (Davoodi et al., 2007), and understanding the neural control of movement (McKay et al., 2007; Berniker et al., 2009). Software for musculoskeletal simulation is widely available today, both from commercial vendors and open source software projects.

Most musculoskeletal simulators are based on algorithms for multibody dynamics initially developed for simulating robots and engineering machinery. At their core, they simulate chains of rigid bodies connected by joints. The rigid bodies represent body segments. Muscles apply forces on the rigid bodies, or torques at the joints (via the muscle's moment arm at the joint), using physiologically reasonable constitutive models. Examples of these types of simulators include software packages such as SIMM (Delp and Loan, 2000), AnyBody (AnyBodyTech, 2009), MSMS (Davoodi et al., 2007), and OpenSim (Delp et al., 2007), and many research simulations, including those used in all the papers cited above.

How should muscle mass be accounted for in these types of models of musculoskeletal dynamics? Muscles comprise the majority of mass in many body segments, such as the human upper arm and thigh, so this is an important decision that any modeler has to make.

One convenient method is to lump the mass of the muscle along with the bones and soft tissues within a body segment; that is, an entire body segment, such as the thigh, is modeled as a single body in the model of dynamics. We will refer to this method as “segment lumping.” This method appears to be widely used in practice. Some authors explicitly state that they use body segment inertial parameters (e.g., Pandy et al., 1990; Kargo et al., 2002; Wu et al., 2008); many others do not specify how they treat muscle mass at all, suggesting that they do not make any special distinction from the body segment mass. Indeed, there is a considerable literature on measuring body segment inertial parameters (BSIPs) and scaling them appropriately (see Jensen, 1993; de Leva, 1996; Dumas et al., 2007, and the extensive references therein).

Surprisingly, there has been little analysis of the effect of this important modeling choice on the dynamics of the musculoskeletal system. Muscles are mechanically coupled—through stiff tendons—to physically distant joints and bones, so one may suspect that the inertias of distal joints will be incorrectly estimated by lumping muscles with their nearest body segments. How large are these errors? Under what circumstances can these errors be ignored and when are they likely to be significant? The present study addresses these questions using models of musculoskeletal systems that are as simple as possible to clearly illustrate the effects of segment lumping.

## 2. Methods

Two mathematical models of simple musculoskeletal systems were constructed and the errors from segment lumping analyzed. These models were deliberately designed to have simple anatomical parameters so that readers can verify the results themselves using pencil and paper calculations. The

E-mail address: [pai@cs.ubc.ca](mailto:pai@cs.ubc.ca)

simplifications are generally conservative (see Appendix B). Generalization to more complex musculoskeletal anatomy is conceptually straightforward but the details may have to be buried in a software implementation which will obscure the essential ideas.

Muscle mass influences the dynamics of the musculoskeletal system in two distinct ways: it serves as a store of kinetic energy during movement, and as a store of potential energy in a gravitational field. The equations of motion are derived from these energies in Lagrangian mechanics and hence errors in modeling energy produce errors in all aspects of dynamics. This paper focuses on kinetic energy since it provides a concise description of the dynamic state of the system and allows for direct comparison of approaches. The contribution of a muscle's mass to the biomechanical system's kinetic energy is calculated for each model, with and without segment lumping. This contribution is completely characterized by the muscle's generalized inertia matrix,  $\mathcal{I}_m$ , to be defined below.

The general approach is to parameterize the kinematics of the musculoskeletal system in terms of a vector of  $n$  joint angles  $\theta = (\theta_1 \dots \theta_n)^T$ , following the usual practice in biomechanics. For simplicity of exposition, muscles and bones are modeled as line segments with uniformly distributed mass along their lengths, and total mass  $\mu$ . Following the standard practice in these types of biomechanical models (e.g., Zajac, 1989; Cheng et al., 2000), the musculotendon is treated as a unit actuator (which I will refer to as "muscle," for short) whose behavior is determined by its length and the time derivatives of length.

As the skeleton moves, the muscle must stretch or shorten between its origin and insertion. The muscle is assumed to be sufficiently active so that it does not buckle. Since the standard musculotendon unit model above ignores variations in physical properties within the muscle, we will assume that it stretches uniformly. The crucial point is that the muscle stretches and shortens during movement, and hence the mass in the muscle must be moved in the direction of the stretch as well, and will contribute to the system's inertia.

The position of a material point  $p$  in a muscle is parameterized by an intrinsic non-dimensional material coordinate  $s \in [0, 1]$  (see Fig. 1);  $s=0$  at muscle origin and  $s=1$  at insertion. The value of  $s$  remains fixed for a material point when the material stretches.

The point's velocity  $\dot{p}$  is related to the system's joint angle velocity  $\dot{\theta}$  using the material point's Jacobian matrix  $J_m(s)$ :

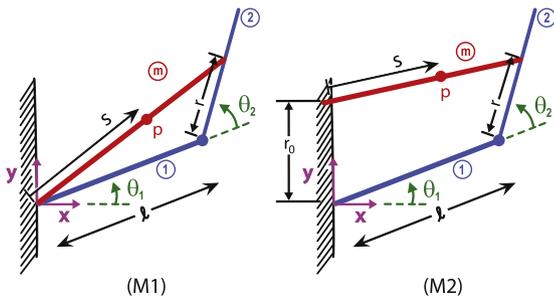
$$\dot{p} \stackrel{\text{def}}{=} J_m(s)\dot{\theta}. \tag{1}$$

Note that this Jacobian gives the three dimensional velocity of the point, and should not be confused with other Jacobians that arise in biomechanics; for instance, the moment arm Jacobian relating the rate of change of muscle length to joint velocity.

The total kinetic energy of a muscle is therefore

$$K_m = \frac{1}{2} \int_0^1 \dot{p}^T \dot{p} \mu_m ds = \frac{1}{2} \dot{\theta}^T \left( \underbrace{\mu_m \int_0^1 J_m(s)^T J_m(s) ds}_{\stackrel{\text{def}}{\mathcal{I}_m}} \right) \dot{\theta}. \tag{2}$$

This computation defines the muscle's generalized inertia  $\mathcal{I}_m$ . It is the contribution of the muscle's mass to the system's inertia, expressed in joint space. Generalized inertia matrices are defined similarly for bones. The total inertia of the system is the sum of these inertia matrices for each muscle and bone in the system.



**Fig. 1.** (M1) Planar musculoskeletal model with a monoarticular muscle labeled  $m$ , two moving bones labeled 1 and 2, and a "grounded" bone. A reference coordinate frame is shown attached to the ground. Bones are connected by revolute joints, with joint angles  $\theta_1$  and  $\theta_2$ , measured as shown. The muscle's origin on bone 1 is assumed, for simplicity, to coincide with the center of the first joint, and its insertion on bone 2 to be offset by a distance  $r$  from second joint. The muscle and bones are line segments with uniformly distributed mass along their lengths, with total mass  $\mu_m, \mu_1$ , and  $\mu_2$ , respectively. We can normalize all lengths by the length of bone 2, without loss of generality, and hence we denote the bone lengths as  $l_1=1$  and  $l_2=1$ . (M2) Biarticular model. This is similar to M1 except that the muscle origin is on the grounded bone, and offset by  $r_0$  from the proximal joint.

### 2.1. M1: a monoarticular model

This is an idealization of the kinematically simplest muscles which span a single joint, for instance, *brachialis* and *soleus*. See Fig. 1(M1).

The coordinates of a material point on the muscle,  $p$ , are given by

$$p = \begin{pmatrix} l \cos \theta_1 + r \cos(\theta_1 + \theta_2) \\ l \sin \theta_1 + r \sin(\theta_1 + \theta_2) \end{pmatrix} s. \tag{3}$$

If the muscle is lumped with the segment of bone 1, the muscle's mass moves with the bone. The material point,  $p^l$ , is still given by Eq. (3), but now  $\theta_2$  is fixed at the angle at which the muscle was lumped (say,  $\pi/2$ ).

### 2.2. M2: a biarticular model

This is a simplification of many common biarticular muscles, for instance, *biceps brachii*, *gastrocnemius*, and *biceps femoris* in mammals. See Fig. 1(M2). The coordinates of a material point on the muscle,  $p$ , are given by

$$p = \begin{pmatrix} 0 \\ r_0 \end{pmatrix} (1-s) + \begin{pmatrix} l \cos \theta_1 + r \cos(\theta_1 + \theta_2) \\ l \sin \theta_1 + r \sin(\theta_1 + \theta_2) \end{pmatrix} s. \tag{4}$$

If, on the other hand, the muscle is lumped with the segment of bone 1 in the configuration  $\theta_1 = 0, \theta_2 = \pi/2$ , the material point,  $p^l$ , is given by

$$p^l = \begin{pmatrix} -r_0 \sin \theta_1 \\ r_0 \cos \theta_1 \end{pmatrix} (1-s) + \begin{pmatrix} l \cos \theta_1 - r \sin \theta_1 \\ l \sin \theta_1 + r \cos \theta_1 \end{pmatrix} s. \tag{5}$$

### 2.3. Numerical simulation

To investigate how inertial errors affect simulations, a specific monoarticular model, based on M1, was created using the following parameters:  $\mu_1 = \mu_2 = \mu_m = 4.20$  kg,  $l_1 = l_2 = 0.43$  m, and  $r = 0.1$ . The parameters were chosen to be approximately the mass and dimensions of the leg of a human male weighing 70 kg, based on Dumas et al. (2007). Half the mass of link 1 (thigh) is assumed to be in the muscle. The intent was not to construct an accurate model of the leg (there is only one muscle in this model) but to pick reasonable values of the parameters. Two numerical experiments were performed with this model; Matlab source code for the experiments is included in the supplementary materials.

Experiment 1 measures trajectory errors due to segment lumping. Equations of motion were derived using the standard Lagrangian formulation (e.g., see Murray et al., 1994, Chapter 4), and numerically integrated over time. The kinetic energy of the system is  $K = \frac{1}{2} \dot{\theta}^T \mathcal{I} \dot{\theta}$  (see Eq. (2)); the potential energy  $V$  due to gravity was modeled, with gravitational acceleration in the  $+x$  direction. Since the goal is to understand the errors in modeling mass, viscoelastic forces in muscle were not included in the simulation. Simulations were performed in Matlab (Mathworks, Natick, MA), using `ode45`, their implementation of the Runge-Kutta(4,5) numerical integrator.

Experiment 2 measures errors in forces and effective inertia at the end point of the limb (see Appendix A).

## 3. Results

### 3.1. M1: a monoarticular model

Since we have an explicit expression for the kinematics of a material point, Eq. (3), the Jacobian matrix  $J_m(s)$  is obtained by differentiating it:

$$\dot{p} = \begin{pmatrix} -l \sin \theta_1 - r \sin(\theta_1 + \theta_2) & -r \sin(\theta_1 + \theta_2) \\ l \cos \theta_1 + r \cos(\theta_1 + \theta_2) & r \cos(\theta_1 + \theta_2) \end{pmatrix} s \dot{\theta} = \underbrace{J_m(s)}_{\stackrel{\text{def}}{\mathcal{I}_m}} \dot{\theta}. \tag{6}$$

Here  $\theta = (\theta_1 \ \theta_2)^T$ , a vector of joint angles. Then the inertia of the muscle (see Eq. (2)) is

$$\mathcal{I}_m = \mu_m \int_0^1 J_m(s)^T J_m(s) ds = \frac{\mu_m}{3} \begin{pmatrix} l^2 + r^2 + 2lr \cos \theta_2 & r^2 + lr \cos \theta_2 \\ r^2 + lr \cos \theta_2 & r^2 \end{pmatrix}. \tag{7}$$

Note (see caption of Fig. 1) that all lengths are normalized by the length of bone 2. Using the same method, the inertias of bones 1 and 2 can be found, and the total inertia of the system

is  $\mathcal{I} = \mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_m$ .

$$\mathcal{I}_1 = \frac{\mu_1}{3} \begin{pmatrix} l^2 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathcal{I}_2 = \frac{\mu_2}{3} \begin{pmatrix} 1 + 3l^2 + 3l\cos\theta_2 & 1 + \frac{3}{2}l\cos\theta_2 \\ 1 + \frac{3}{2}l\cos\theta_2 & 1 \end{pmatrix}. \quad (8)$$

If the muscle was lumped with bone 1 (with the insertion fixed at  $\theta_2 = \pi/2$ ) the muscle's inertia matrix becomes<sup>1</sup>

$$\mathcal{I}_m^L = \frac{\mu_m}{3} \begin{pmatrix} l^2 + r^2 & 0 \\ 0 & 0 \end{pmatrix}. \quad (9)$$

Comparing  $\mathcal{I}$  and  $\mathcal{I}^L$ , we see that the error in inertia is

$$\mathcal{E} = \mathcal{I} - \mathcal{I}^L = \mathcal{I}_m - \mathcal{I}_m^L = \frac{\mu_m}{3} \begin{pmatrix} 2l\cos\theta_2 & r^2 + l\cos\theta_2 \\ r^2 + l\cos\theta_2 & r^2 \end{pmatrix}. \quad (10)$$

One can make several observations about the error. The magnitude depends on the configuration of the body. There are significant off-diagonal terms which means that the error couples velocities of different joints, making it more difficult to remove the error by calibration.

The error decreases with the insertion distance  $r$ . However, in physiological conditions one cannot hope that  $r$  will be very small, since this would also reduce the moment arm, and hence the torque on the joint. Specifically, the moment arm of the muscle at joint 2 is  $\alpha_2 = l\sin\theta_2 / \sqrt{l^2 + r^2 + 2lr\cos\theta_2}$ . As  $r \rightarrow 0$ , both the inertia error  $\mathcal{E}$  and the moment arm  $\alpha_2$  asymptotically decrease linearly with  $r$ .

The error decreases in some postures.  $\mathcal{E}(1,1)$  is reduced to zero when  $\theta_2 = \pi/2$ . But all other elements of the matrix will still have an error of  $\mu_m/3 \cdot r^2$ .

It is instructive to look at  $\mathcal{E}(2,2)$ , the error in self-inertia<sup>2</sup> experienced during motions that only involve the second joint, which is independent of the posture of the joint. This error, relative to the segment lumped inertia  $\mathcal{I}^L$  used in current models, is

$$\varepsilon_{2,2} \stackrel{\text{def}}{=} \frac{\mathcal{E}(2,2)}{\mathcal{I}^L(2,2)} = \frac{\mu_m}{\mu_2} r^2. \quad (11)$$

Thus the error can be significant if  $r$  is large, and if the mass of the muscle is large relative to the mass of the segment on which it inserts.

### 3.2. Consequences of inertial errors for simulation

Errors in inertia affect all aspects of a dynamic simulation. In the first experiment, the monoarticular leg model (Section 2.3) was simulated, starting with the knee raised and dropping passively to the ground. See Fig. 2. The two models, with and without lumping, quickly diverge, with a 14° error in  $\theta_2$  within 300 ms. This also results in errors in “foot” position; when the unlumped foot reaches the ground, the lumped foot is still 2.9 cm above the ground. The total energy remains essentially constant during simulation in both the lumped and unlumped models, showing that energy is conserved in both. The solutions diverge only because of the differences in inertia. Thus we see that errors in inertia produce errors that accumulate over time, rather quickly.

In a second experiment errors in the effective inertia of the endpoint of the limb were computed and found to be large in natural configurations (see Appendix A for details).

### 3.3. M2: a biarticular model

Once again, we can simply differentiate the expression for the kinematics of a material point, Eq. (4), to obtain the Jacobian matrix  $J_m(s)$ . It turns out to be the same expression (Eq. (6)) we obtained for example M1; this is not surprising, since in M1 the muscle origin is assumed to be at the center of the first joint, and hence also has zero velocity, just like M2. Indeed, M1 is a special case of M2, with the origin offset,  $r_o$ , set to zero. Thus  $\mathcal{I}_m$  is also the same as in our example M1 and is given by Eq. (7).

The main difference is that when the muscle is lumped with bone 1 the muscle's inertia matrix becomes a bit more complicated. If segment lumping was performed in the posture  $\theta_1 = 0$  and  $\theta_2 = \pi/2$ ,

$$\mathcal{I}_m^L = \frac{\mu_m}{3} \begin{pmatrix} l^2 + r^2 + r_o(r + r_o) & 0 \\ 0 & 0 \end{pmatrix}. \quad (12)$$

When  $r_o \rightarrow 0$ , Eq. (12) becomes the same as Eq. (9), as one would expect. When  $r_o \rightarrow r$ ,  $\mathcal{I}_m^L(1,1) \rightarrow \mu_m(l^2/3 + r_o^2)$ , as one would predict from the parallel axis theorem.

Comparing  $\mathcal{I}$  and  $\mathcal{I}^L$ , we see that the error in inertia is now

$$\mathcal{I}_m - \mathcal{I}_m^L = \frac{\mu_m}{3} \begin{pmatrix} 2l\cos\theta_2 - r_o(r + r_o) & r^2 + l\cos\theta_2 \\ r^2 + l\cos\theta_2 & r^2 \end{pmatrix}. \quad (13)$$

Interestingly, the error in the  $\mathcal{I}(2,2)$  element, the inertia experienced during motions that only involve the second joint, is the same as in the M1 case (Eq. (11)). Therefore, monoarticular and biarticular muscles can be combined when estimating this important error.

### 3.4. Human triceps surae

A simplified model of the *triceps surae* of the human ankle, comprising *gastrocnemius* and *soleus*, is shown in Fig. 3. We will compare the contributions to  $\mathcal{I}(2,2)$ , the self-inertia of ankle joint, due to the mass of the foot and triceps surae.

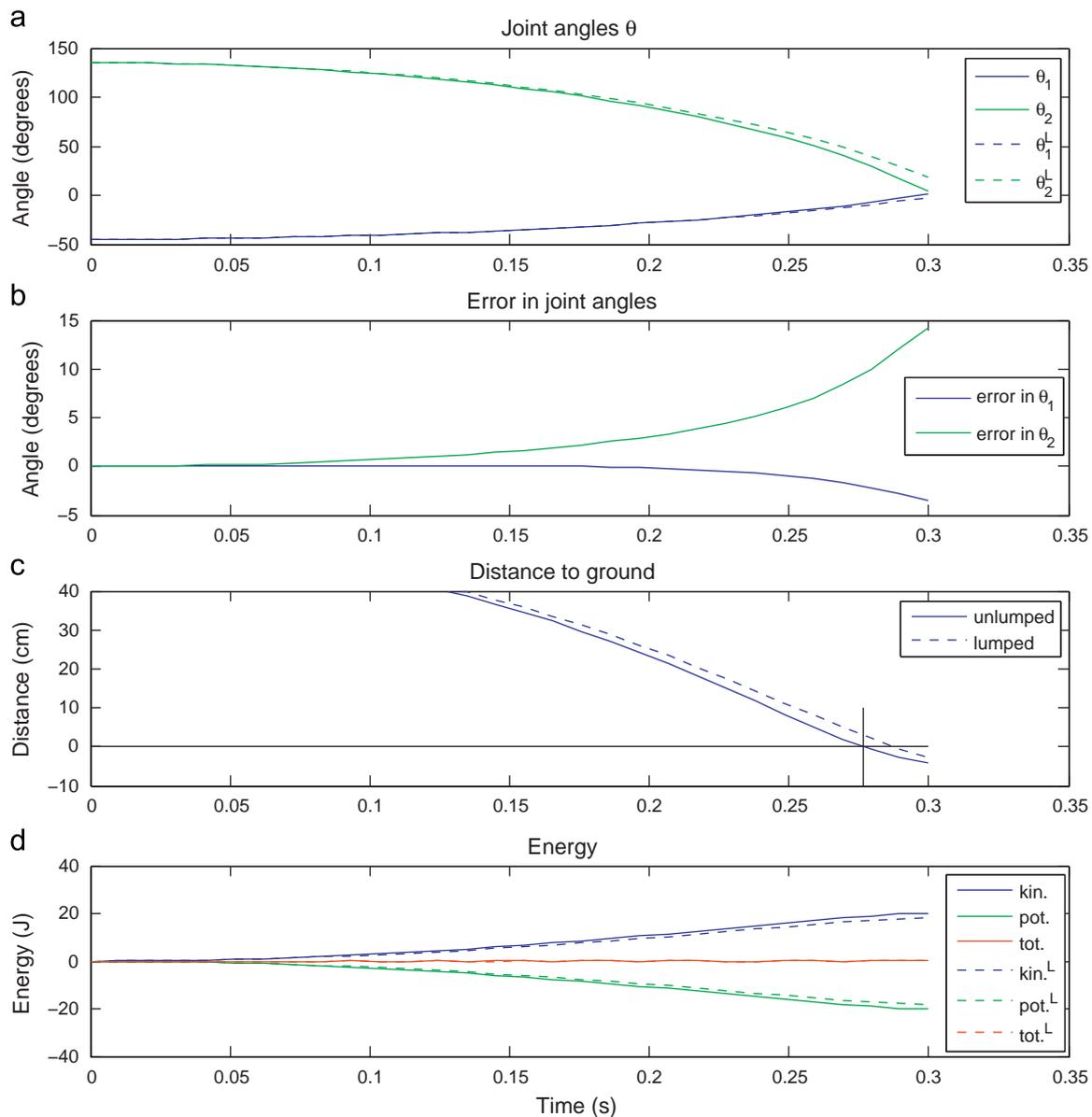
Based on Sections 3.1 and 3.3, the formulae are identical for the monoarticular *soleus* and the biarticular *gastrocnemius*. The masses of the two muscles were estimated based on Wickiewicz et al. (1983) to be 0.215 and 0.158 kg, respectively, for a muscle mass of  $\mu_m = 0.373$  kg. The insertion distance from the ankle joint was estimated from Binder-MacLeod et al. (2009) to be  $r = 42.7$  mm. This is the average measured moment arm when the foot is plantarflexed by 10°. This angle was chosen to maximize the moment arm, which is a conservative estimate since  $r$  is always greater than the moment arm at any angle. With these values,  $\mathcal{I}_m(2,2) = 0.227 \times 10^{-3} \text{ kg m}^2$ .

The inertia of the foot about the ankle was computed from anthropometric data in Dumas et al. (2007, Table 2). The values were averaged for males and females, since the sex and body segment dimensions of the subject were not reported in Wickiewicz et al. (1983). Scaled to a total body mass of 70 kg and an average foot length of 0.249 m, we estimate  $\mathcal{I}_f(2,2) = 2.98 \times 10^{-3} \text{ kg m}^2$ .

Therefore just the triceps surae (which account for about half the muscle mass in the shank) contribute an additional 7.6% to the self-inertia of the ankle joint, which is erroneously ignored if the muscle mass is lumped with the shank segment. These numbers are approximate as anthropometric data were combined from different sources and subjects, but it is clear that such errors due to segment lumping are quite large.

<sup>1</sup> Substitute  $\theta_2 = \pi/2$  in Eq. (3) and recompute  $\mathcal{I}_m$ .

<sup>2</sup> The “self-inertia” of joint  $n$  is  $\mathcal{I}(n,n)$ : the inertia experienced if you move just that joint.



**Fig. 2.** Comparison of movement with and without lumping. The limb was simulated for 300 ms, starting at rest ( $\dot{\theta} = 0$ ) with joint angles  $\theta_1 = -\pi/4, \theta_2 = 3\pi/4$ . (a) Joint angles with ( $\theta^l$ , dashed lines) and without ( $\theta$ , solid lines) lumping. (b) The error in joint angles,  $\theta^l - \theta$ . (c) Distance of the end point (foot) from the ground plane which was 0.817 m (i.e.,  $1.9 \times l_2$ ) below the first joint, with (dashed lines) and without (solid lines) lumping. The short vertical line marks the time at which the unlumped end point reaches the ground. At that time, the lumped model's end point is still 2.9 cm above ground. (d) Energy variation during simulation, to verify that the simulation is physically consistent. Blue, kinetic energy; green, potential energy; and red, total energy (kinetic+potential). Potential energy is assumed to be zero at the initial configuration, without loss of generality. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 3.5. Rat biceps femoris posterior

Biceps femoris posterior (BFp) is a broad biarticular muscle that inserts on the tibia of the rat hind limb and plays an important role in locomotion. See Fig. 4. It is instructive to compare the contributions to self-inertia of the knee joint from the BFp and the bones of the shank. This is similar to the computation in Eq. (11), with the shank segment replaced by the shank bones. From the measurements of Johnson et al. (2008),  $r=0.42$ . The relevant masses were measured by dissecting an adult female Sprague–Dawley rat (300 g) in the laboratory of Prof. Matthew Tresch, using protocols approved by the Northwestern University ACUC. Tibia and fibula together weighed  $\mu_{tibfib} = 0.801$  g, and BFp had a wet weight of  $\mu_{BFp} = 2.046$  g (a number which is consistent with the published results of Eng

et al., 2008, who measured both heads of BF together). Then

$$\frac{\mathcal{I}_{BFp}(2,2)}{\mathcal{I}_{tibfib}(2,2)} = \frac{\mu_{BFp}r^2}{\mu_{tibfib}} = 0.45. \quad (14)$$

Therefore, ignoring BFp's contribution to the self-inertia of the knee as a consequence of segment lumping is comparable to ignoring 45% of the mass of the bones of the shank.

## 4. Discussion

The preceding analyses and numerical examples show that errors in inertia due to the common practice of segment lumping can be quite large and variable, changing with body posture and coupling the velocities of different joints. The errors can be

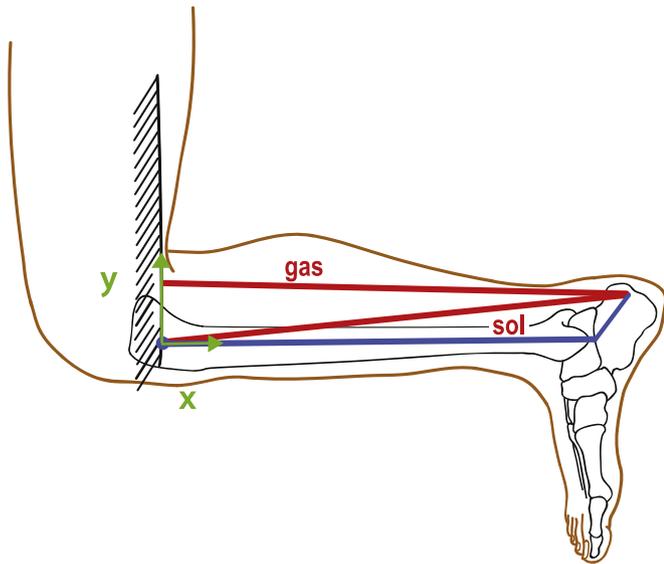


Fig. 3. Human triceps surae.

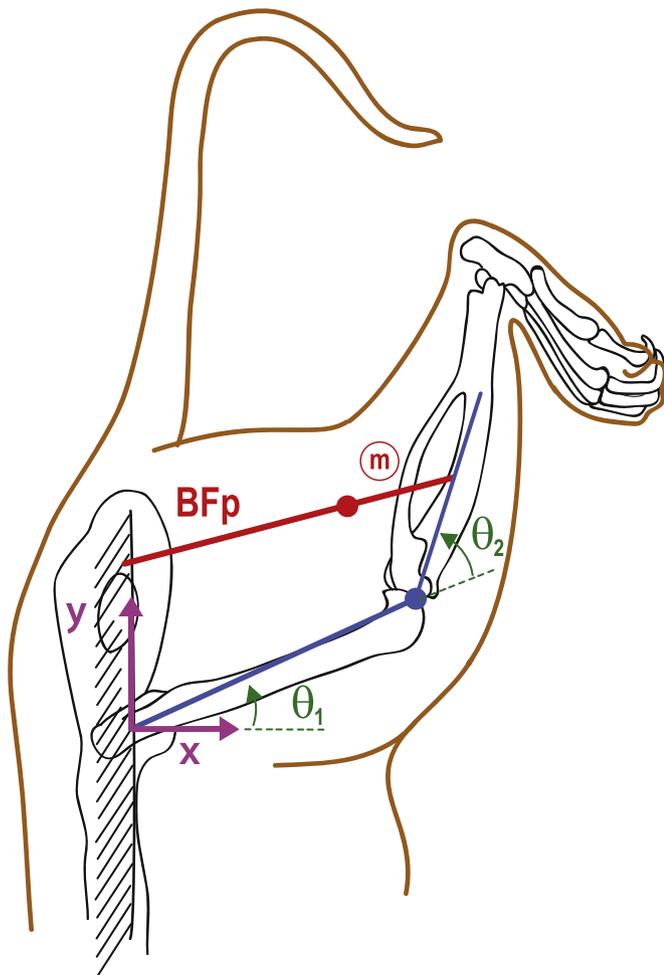


Fig. 4. Rat biceps femoris posterior.

reduced in some postures but cannot be easily eliminated. The examples suggest that even the smallest errors can be quite large. The errors are likely to remain even if more details, such as soft tissue and muscle wobble, are included (Appendices B and C),

since the errors are due to the fundamental inertial coupling of a muscle to distal joints.

Errors in inertia lead to errors in all aspects of a dynamic simulation. They directly translate into errors in predicted muscle forces and computed accelerations (which are integrated over time and hence accumulate). Inertia errors are likely to have a significant effect in highly dynamic simulations involving impacts and friction, such as tapping with a finger, throwing a ball, or kicking with a toe. For impacts with impulsive loads, such as during running or landing on one's feet, errors in inertia will cause proportional errors in the velocity change after impact.

What can be done about these errors?

One option is to do nothing. This has the advantage of convenience, since both software and body segment inertial parameters (BSIPs) are easily available. Some applications may have high error tolerance; indeed, dynamic simulations are sometimes used to answer essentially kinematic or static questions. Questions about muscle moment arms and isometric forces are of this type, and may not be affected by errors in dynamic simulation.

A second option is to explicitly include muscle mass within a musculotendon unit model. Most current models of musculotendon units are massless (Zajac, 1989). Some models include a muscle mass parameter but use it only for stabilizing the internal dynamics of the contractile element (e.g., Virtual Muscle, Cheng et al., 2000, uses mass in this way and to scale the physiological cross-sectional area of the muscle). To contribute to the musculoskeletal system's dynamics, the acceleration of the muscle mass must be coupled with the acceleration of the body segments and must not be solved for independently. This will require a non-trivial change to existing simulators, similar to the inclusion of closed kinematic loops. Another challenge is that individual muscle masses are now required for simulation, rather than just BSIPs.

A third option is to compute the full contribution of each muscle to the joint space inertia matrix,  $\mathcal{I}_m$ , as suggested in this paper (Eqs. (2) and (7)). This has the advantage of retaining current models of force generation in muscle while accounting better for the muscle's mass. In principle, existing simulators could be revised to incorporate  $\mathcal{I}_m$  fully. One challenge is how to adapt the fast linear time algorithms developed for serial multi-body chains, while retaining their efficiency. It may be possible to design such efficient algorithms, as many previous fast algorithms are implicitly based on exploiting matrix sparsity (Pai et al., 2000). This also requires individual muscle masses.

A final option is to use muscle models based on continuum mechanics (for instance, finite element models (FEM), e.g., Blemker and Delp, 2005, or strands, Sueda et al., 2008). These models represent muscle mass at nodal points within a muscle, and therefore account for not only the total mass of the muscle but also for changes in mass distribution within muscle during movement. FEM models with volumetric elements are very general but can have significant computational cost and modeling complexity. Fully dynamic FEM simulations have therefore been limited to small numbers of muscles and joints. Strand models use one dimensional elements that can curve in space and represent the dynamics of a set of fibers, such as muscle, tendon, or fascicle. They are designed for large scale musculoskeletal simulation for motor control, with many constraints such as tendon sheaths. A challenge for both models is that they need the distribution of muscle mass and material properties within muscle.

Perhaps the most important observation here is that a muscle's mass can influence the self-inertia of joints that are physically distant and distal to it. This is impossible in kinematic chains of rigid bodies that are widely used in current biomechanical models; the mass of a proximal body has no effect on the

self-inertia of a distal joint in those models. The analysis presented here shows, for example, that the self-inertia of the distal interphalangeal joint of the finger is influenced not only by the mass of the tiny distal phalanx, but also of the larger extrinsic muscles in the forearm. This may be a useful physical principle exploited by nature for dealing with large and impulsive loads on the extremities.

### Conflict of interest statement

The author does not have a conflict of interest regarding this manuscript.

### Acknowledgments

The author would like to thank S. Sueda, M. Tresch, A. Ruina, and S.-H. Yeo for useful feedback on the manuscript, and M. Tresch and M. Wu for providing data on the rat hindlimb. This work was supported in part by NIH Grant R01AR053608 and a CIHR (CRCNS) operating grant. Additional support for the laboratory was provided by the Peter Wall Institute for Advanced Studies, the Human Frontier Science Program, NSERC, and the Canada Research Chairs Program. These agencies had no involvement in the design or implementation of the study.

### Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at doi:[10.1016/j.jbiomech.2010.04.004](https://doi.org/10.1016/j.jbiomech.2010.04.004).

### References

- AnyBodyTech, 2009. Anybody Modeling System. Software package <<http://www.anybodytech.com/>>.
- Berniker, M., Jarc, A., Bizzi, E., Tresch, M.C., 2009. Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics. *Proc. Natl. Acad. Sci. USA* 106, 7601–7606.
- Binder-MacLeod, B.I., Manal, K., Buchanan, T.S., 2009. A novel approach for experimental derived muscle parameters of the soleus muscle. In: American Society of Biomechanics, 2009 Annual Meeting Conference Abstract No. 1047 <<http://www.asbweb.org/conferences/2009/pdf/1047.pdf>>.
- Blemker, S.S., Delp, S.L., 2005. Three-dimensional representation of complex muscle architectures and geometries. *Ann. Biomedical Eng.* 33 (5), 661–673.
- Cheng, E., Brown, I., Loeb, G., 2000. Virtual muscle: a computational approach to understanding the effects of muscle properties on motor control. *J. Neurosci. Methods* 101, 117–130.
- Davoodi, R., Urata, C., Hauschild, M., Khachani, M., Loeb, G.E., 2007. Model-based development of neural prostheses for movement. *IEEE Trans. Biomed. Eng.* 54, 1909–1918.
- de Leva, P., 1996. Adjustments to Zatsiorsky–Seluyanov's segment inertia parameters. *J. Biomech.* 29, 1223–1230.
- Delp, S., Loan, J., 2000. A computational framework for simulating and analyzing human and animal movement. *Comput. Sci. Eng.* 2, 46–55.
- Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., Thelen, D.G., 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* 54, 1940–1950.
- Dumas, R., Chèze, L., Verriest, J.P., 2007. Adjustments to McConville et al. and Young et al. body segment inertial parameters. *J. Biomech.* 40, 543–553.
- Eng, C.M., Smallwood, L.H., Rainiero, M.P., Lahey, M., Ward, S.R., Lieber, R.L., 2008. Scaling of muscle architecture and fiber types in the rat hindlimb. *J. Exp. Biol.* 211, 2336–2345.
- Hollerbach, J.M., Flash, T., 1982. Dynamic interactions between limb segments during planar arm movement. *Biol. Cybern.* 44, 67–77.
- Jensen, R.K., 1993. Human morphology: its role in the mechanics of movement. *J. Biomech.* 26 (Suppl. 1), 81–94.
- Johnson, W.L., Jindrich, D.L., Roy, R.R., Reggie Edgerton, V., 2008. A three-dimensional model of the rat hindlimb: musculoskeletal geometry and muscle moment arms. *J. Biomech.* 41, 610–619.
- Kargo, W., Nelson, F., Rome, L., 2002. Jumping in frogs: assessing the design of the skeletal system by anatomically realistic modeling and forward dynamic simulation. *J. Exp. Biol.* 205 (Pt 12), 1683–1702.
- McKay, J.L., Burkholder, T.J., Ting, L.H., 2007. Biomechanical capabilities influence postural control strategies in the cat hindlimb. *J. Biomech.* 40, 2254–2260.
- Murray, R., Li, Z., Sastry, S.S., 1994. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL.
- Pai, D.K., Ascher, U.M., Kry, P.G., 2000. Forward dynamics algorithms for multibody chains and contact. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp. 857–863.
- Pandy, M.G., Zajac, F.E., Sim, E., Levine, W.S., 1990. An optimal control model for maximum-height human jumping. *J. Biomech.* 23, 1185–1198.
- Sueda, S., Kaufman, A., Pai, D.K., 2008. Musculotendon simulation for hand animation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 27 (3), 83:1–83:8.
- Wickiewicz, T.L., Roy, R.R., Powell, P.L., Edgerton, V.R., 1983. Muscle architecture of the human lower limb. *Clin. Orthop. Relat. Res.* 275–283.
- Wu, J.Z., An, K.N., Cutlip, R.G., Krajnak, K., Welcome, D., Dong, R.G., 2008. Analysis of musculoskeletal loading in an index finger during tapping. *J. Biomech.* 41, 668–676.
- Zajac, F.E., 1989. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *CRC Crit. Rev. Biomedical Eng.* 17 (4), 359–411.