

Generating dynamic simulations of movement using computed muscle control

Darryl G. Thelen^a, Frank C. Anderson^{b,*}, Scott L. Delp^b

^a *Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706-1572, USA*

^b *Department of Mechanical Engineering, Biomechanical Engineering Division, Stanford University, Stanford, CA 94305 4038, USA*

Accepted 12 November 2002

Abstract

Computation of muscle excitation patterns that produce coordinated movements of muscle-actuated dynamic models is an important and challenging problem. Using dynamic optimization to compute excitation patterns comes at a large computational cost, which has limited the use of muscle-actuated simulations. This paper introduces a new algorithm, which we call computed muscle control, that uses static optimization along with feedforward and feedback controls to drive the kinematic trajectory of a musculoskeletal model toward a set of desired kinematics. We illustrate the algorithm by computing a set of muscle excitations that drive a 30-muscle, 3-degree-of-freedom model of pedaling to track measured pedaling kinematics and forces. Only 10 min of computer time were required to compute muscle excitations that reproduced the measured pedaling dynamics, which is over two orders of magnitude faster than conventional dynamic optimization techniques. Simulated kinematics were within 1° of experimental values, simulated pedal forces were within one standard deviation of measured pedal forces for nearly all of the crank cycle, and computed muscle excitations were similar in timing to measured electromyographic patterns. The speed and accuracy of this new algorithm improves the feasibility of using detailed musculoskeletal models to simulate and analyze movement.

© 2003 Elsevier Science Ltd. All rights reserved.

Keywords: Musculoskeletal modeling; Dynamic simulation; Optimization; Control; Pedaling

1. Introduction

Dynamic simulation is a powerful approach for investigating how the elements of the neuromusculoskeletal system interact to produce movement. Forward dynamic simulation provides capabilities not generally offered by experimental approaches. For example, muscle excitation patterns or other parameters of a model can be altered to determine how they affect movement. This type of analysis has been used to study neural control of movement (Zajac, 1993), design neuromuscular stimulation systems (Yamaguchi and Zajac, 1990), evaluate the causes of pathological movement (Riley and Kerrigan, 1998), and design prosthetic devices (Piazza and Delp, 2001).

Determining a set of muscle excitations that produce a desired movement is one of the major challenges in

creating a forward dynamic simulation. One approach is to use dynamic optimization to determine a set of muscle excitations that generate a simulation that best reproduces experimental data (Davy and Audu, 1987; Yamaguchi and Zajac, 1990; Neptune and Hull, 1998; Kaplan and Heegaard, 2001). Using this approach, the optimization objective function is typically a global measure of the error between measured and simulated biomechanical quantities. Solving dynamic optimization problems can be computationally overwhelming when a model includes many muscles and allows for complex excitation patterns. Consequently, almost all previous studies have reduced the number of muscles (Davy and Audu, 1987; Yamaguchi and Zajac, 1990) or simplified the muscle control signals (e.g., by assuming block excitation patterns; Neptune and Hull, 1998) to minimize the number of variables included in the optimization problem. Even with these simplifications, solving dynamic optimization problems can require thousands of complete integrations of the model state equations.

*Corresponding author. Tel.: +1-650-736-0801; fax: +1-650-725-1587.

E-mail address: fca@stanford.edu (F.C. Anderson).

Two novel techniques have been described that reduce the computational demands of standard dynamic optimization techniques. Yamaguchi et al. (1995) used a pseudo-inverse method to efficiently compute muscle forces that generate desired joint accelerations at each time step of a forward dynamic simulation. However, their approach does not incorporate the dynamic properties of muscles and requires the use of a specific optimization function to resolve muscle redundancy. Kaplan and Heegaard (2001) used a second-order dynamic optimization technique based on discretizing the state equations. Although this technique greatly improved performance, it is difficult to implement because it requires one to symbolically formulate first and second derivatives of the state equations with respect to the control variables.

There is a need for new techniques to generate coordinated forward dynamic simulations that reduce the computational burden of dynamic optimization, rely less on simplifying assumptions, and are straightforward to implement. This paper introduces a new method, which we term “computed muscle control”, for determining a set of muscle excitations that drives a muscle-actuated model to track experimental data. Computed muscle control reduces the computational burden by requiring only one integration of the state equations. It allows one to include many muscles in a model without incurring large computational costs and produces continuously varying muscle excitations that more realistically represent physiologic excitation patterns. We demonstrate the utility of computed muscle control by producing a muscle-actuated forward simulation that reproduces measured pedaling dynamics.

2. Methods

We first describe the basic elements of a forward dynamic musculoskeletal model. We then introduce a general formulation of the computed muscle control algorithm and describe how it was used to produce a muscle-actuated simulation of pedaling.

2.1. Elements of forward dynamic simulation

A forward dynamic simulation is performed by integrating a set of ordinary differential equations (i.e., state equations) that describe the properties of the musculoskeletal system and the interactions with the environment. Although there are potentially many forms for these equations, they typically include first-order equations for activation dynamics, first-order equations for musculotendon contraction dynamics, and second-order equations of motion for the body.

Activation dynamics (i.e., the process by which muscle-fiber calcium concentration is modulated by

motor unit action potentials) can be modeled by relating the time rate of change of muscle activation (\dot{a}) to muscle activation (a) and excitation (u):

$$\dot{a} = \begin{cases} (u - a) \cdot [u/\tau_{\text{act}} + (1 - u)/\tau_{\text{deact}}], & u \geq a, \\ (u - a)/\tau_{\text{deact}}, & u < a, \end{cases} \quad (1)$$

where τ_{act} and τ_{deact} are the time constants for activation and deactivation, respectively (Raasch et al., 1997). Excitation and activation levels are allowed to vary continuously between zero (no excitation and activation) and one (full excitation and activation).

Musculotendon contraction dynamics can be described by a lumped-parameter model that accounts for the force-length-velocity properties of muscle and the elastic properties of tendon (Zajac, 1989). In particular, the time rate of change of muscle length (\dot{l}_m) can be related to muscle length (l_m), musculotendon length (l_{mt}), and muscle activation (a):

$$\dot{l}_m = f_v^{-1}(l_m, l_{mt}, a), \quad (2)$$

where f_v is the force velocity relation for muscle (Schutte et al., 1993).

The accelerations of the generalized coordinates of the model in response to applied forces can be computed by solving the equations of motion:

$$\ddot{\vec{q}} = \vec{A}^{-1}(\vec{q}) \cdot \{ \vec{G}(\vec{q}) + \vec{C}(\vec{q}, \dot{\vec{q}}) + \vec{R}(\vec{q}) \cdot \vec{f}_m + \vec{E}(\vec{q}, \dot{\vec{q}}) \}, \quad (3)$$

where \vec{q} , $\dot{\vec{q}}$, and $\ddot{\vec{q}}$ are the generalized coordinates, speeds, and accelerations of the model, respectively, \vec{A}^{-1} is the inverse of the system mass matrix, \vec{G} is a vector of generalized forces arising from gravity, \vec{C} is a vector of generalized forces arising from Coriolis and centripetal forces, \vec{R} is a matrix of muscle moment arms, \vec{f}_m is a vector of muscle forces, and \vec{E} is a vector of generalized forces that characterizes the interactions with the environment.

To simulate motion based on a set of input muscle excitations, state Eqs. (1)–(3) are numerically integrated forward in time starting from a set of initial states to produce the time histories of muscle activations, muscle lengths, generalized speeds, and generalized coordinates.

2.2. Computed muscle control algorithm

Our approach is an extension of computed torque control (Lewis et al., 1993), in which feedforward and feedback control is used to drive the kinematic trajectory of a dynamic model toward a set of experimental kinematics. Computed muscle control extends computed torque control by resolving actuator redundancy and accounting for the dynamic force-generating properties of musculotendon actuators.

The computed muscle control algorithm is applied at each integration time step during a forward dynamic

simulation and is comprised of four stages (Fig. 1). In Stage 1, a set of desired accelerations (\ddot{q}_d) is computed based on a set of experimental kinematics and the current kinematic state of the model:

$$\ddot{q}_d = \ddot{q}_{exp} + k_v \underbrace{(\dot{q}_{exp} - \dot{q})}_{\dot{e}_q} + k_p \underbrace{(q_{exp} - q)}_{e_q}, \quad (4)$$

where \dot{q} and q are the generalized speeds and coordinates of the model, \ddot{q}_{exp} , \dot{q}_{exp} , and q_{exp} are the experimental accelerations, velocities, and positions corresponding to the generalized coordinates of the model, and k_v and k_p are feedback gains for the velocity errors (\dot{e}_q) and position errors (e_q), respectively.

If the desired accelerations as given by Eq. (4) are achieved, the velocity and position errors will be driven to zero and display behavior described by a set of decoupled second-order ordinary differential equations (Lewis et al., 1993)

$$\ddot{e}_q + k_v \dot{e}_q + k_p e_q = 0, \quad (5)$$

where \ddot{e}_q are the acceleration errors (i.e., $\ddot{q}_{exp} - \ddot{q}_d$). It is therefore possible to choose the feedback gains so that the errors fall to zero in some desired fashion. For example, if $k_v = 2\sqrt{k_p}$ then the errors will fall to zero in a critically damped manner.

In Stage 2, an optimization problem is solved to compute a set of muscle activations (\bar{a}^*) that give rise to muscle forces that, under steady-state conditions, produce the desired accelerations \ddot{q}_d computed in Stage 1. To estimate the accelerations that result from a set of activations, steady-state muscle forces (\bar{f}_m^*) are computed from the activations by accounting for the force-length-velocity properties of muscle and assuming contraction dynamics have equilibrated. Then, these steady-state forces are applied to the model, and the

equations of motion are solved for the accelerations

$$\ddot{q}^* = \overleftarrow{A}^{-1}(\overrightarrow{q}) \cdot \left\{ \overrightarrow{G}(\overrightarrow{q}) + \overrightarrow{C}(\overrightarrow{q}) \cdot \dot{\overrightarrow{q}}^* + \overrightarrow{R}(\overrightarrow{q}) \cdot \overrightarrow{f}_m^* + \overrightarrow{E}(\overrightarrow{q}, \dot{\overrightarrow{q}}^*) \right\}. \quad (6)$$

The asterisks on \bar{a}^* , \bar{f}_m^* , \ddot{q}^* are used to distinguish these steady-state quantities from the corresponding quantities (\bar{a} , f_m , \ddot{q}) that actually occur during the forward dynamic simulation.

There is some flexibility in the formulation of the optimization problem, but two basic requirements must be met. The first requirement is resolving actuator redundancy by posing a performance criterion (Crowninshield and Brand, 1981; Dul et al., 1984; Kaufman et al., 1991). The second requirement is finding a set of activations that generate the desired accelerations. This can be accomplished by enforcing constraints of the form $\ddot{q}_d - \ddot{q}^* = 0$. It is also possible to control other aspects of the simulation by incorporating additional equality or inequality constraints. For example, one can limit the desired activations based on recorded EMG activity and/or track force measurements.

In Stage 3, a linear proportional feedback controller is used to compute excitations that drive the muscle activations (\bar{a}) to track \bar{a}^* :

$$\bar{u} = \bar{a}^* + k_u(\bar{a}^* - \bar{a}). \quad (7)$$

Here, k_u is a feedback gain, \bar{a}^* is the vector of activations computed in Stage 2, and \bar{a} is the vector of current activations in the forward dynamic simulation. If any muscle excitation values generated by Eq. (7) are below 0.0 or above 1.0, they are limited to 0.0 or 1.0, respectively.

In Stage 4, the muscle excitations are input into the forward dynamic model and numerical integration is

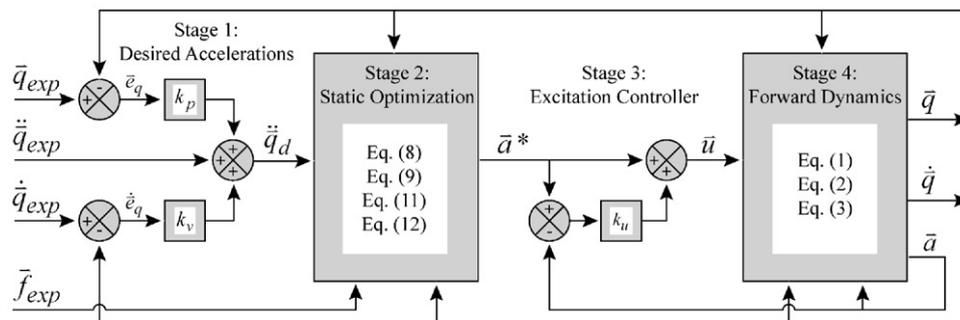


Fig. 1. Schematic of the computed muscle control algorithm. The algorithm is applied at each integration time step of a forward dynamic simulation. In Stage 1, a set of desired accelerations \ddot{q}_d is computed that will drive the generalized coordinates and speeds of the model (q and \dot{q}) toward the experimental kinematics (q_{exp} and \dot{q}_{exp}). The experimental accelerations \ddot{q}_{exp} are included as a feedforward term. The positive constants k_v and k_p are feedback gains for the velocity errors (\dot{e}_q) and position errors (e_q), respectively. In Stage 2, a set of muscle activations (\bar{a}^*) is found using static optimization; under steady-state conditions, these activations produce muscle forces that generate the desired accelerations computed in Stage 1. Additional experimental data, such as external forces (\bar{f}_{exp}), can also be incorporated into the static optimization problem. In Stage 3, a set of neural excitations (\bar{u}) is computed from the muscle activations using a linear controller with a feedback gain of k_u . In Stage 4, the neural excitations are input to the forward dynamic model and numerical integration is used to advance the states to the next time step. The generalized coordinates and speeds of the model are then fed back, and the tracking algorithm is applied repeatedly until the simulation runs to completion.

used to advance the states to the next time step. Note that only the muscle excitations (\bar{u}) computed in Stage 3 are used as input to Stage 4; the steady-state muscle activations (\bar{a}^*) and forces (\bar{f}_m^*) are not used in Stage 4. Rather, the muscle activations (\bar{a}) and forces (\bar{f}_m) generated by the forward dynamic simulation are governed by state Eqs. (1) and (2) and, thus, activation and contraction dynamics are included. The generalized coordinates and speeds are then fed back, and the tracking algorithm is applied repeatedly until the simulation runs to completion.

2.3. Implementation for pedaling

To illustrate the application of the computed muscle control algorithm and to assess its computational performance, we applied the algorithm to pedaling. A planar two-legged forward dynamics model of pedaling was used (Fregly, 1993; Fregly and Zajac, 1996; Raasch et al., 1997; Neptune and Hull, 1998) (Fig. 2). The model was actuated by 30 muscles and included nine segments: a pelvis and a left and right femur, tibia, patella, and foot. Hip and ankle joints were modeled as revolute. A planar kinematic joint that allowed sliding-rolling between the femur, tibia, and patella was used to represent each knee (Delp et al., 1990). The pelvis was fixed, the crank was pinned to ground, pedals were pinned to the crank, and the feet were rigidly connected to the pedals. Due to the closed loops between the pelvis

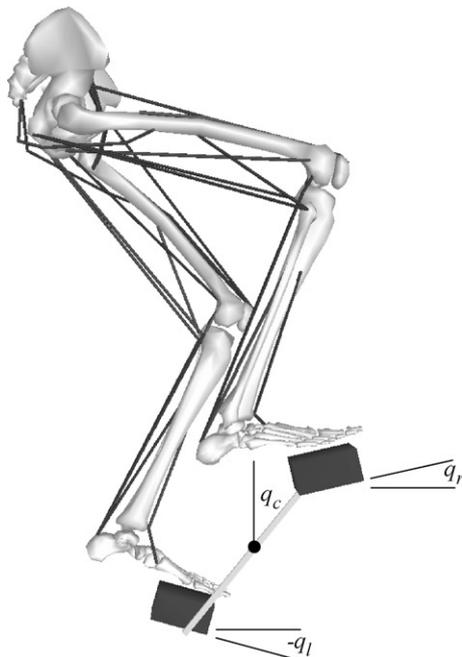


Fig. 2. A planar two-legged model was used to simulate pedaling. The model has three independent degrees of freedom. Crank (q_c) and right and left pedal angles (q_r, q_l) were selected as the independent generalized coordinates. Fifteen muscle–tendon units, each with an independent excitation signal, actuated each leg.

and crank, the model had only three degrees of freedom. The crank angle and left and right pedal angles were used as the independent generalized coordinates: q_c , q_r , and q_l , respectively. Musculoskeletal geometry and parameters were based on the lower extremity model developed by Delp et al. (1990). The optimal fiber lengthened of soleus was lengthened by 0.02 m beyond that reported in Delp et al. (1990) to improve the isometric torque–angle curve for this muscle, as was done in Anderson and Pandey (1999). Muscle activation dynamics was modeled by Eq. (1) with activation and deactivation time constants of 15 ms and 50 ms, respectively. Muscle contraction dynamics were represented by Eq. (2). A fourth order, variable-step Runge–Kutta integrator was used to integrate the state equations.

Average crank angles and pedal forces of ten male competitive cyclists from Neptune et al. (1997) were used as experimental input to the computed muscle control algorithm. The cyclists in that study rode at a work rate of 250 W and crank frequency of 60 RPM. The time histories of the crank and pedal angles were fit using quintic splines and differentiated to form analytical expressions for \dot{q}_{exp} , \ddot{q}_{exp} , and \dddot{q}_{exp} . Crankload dynamics were modeled by an equivalent inertial and resistive torque applied about the center of the crank arm (Fregly, 1993).

For the static optimization problem, the performance criterion (J) was chosen to be the sum of squared muscle activations (Anderson and Pandey, 2001):

$$J = \sum_{m=1}^{30} (a_m^*)^2, \quad (8)$$

where a_m^* is the steady-state activation of muscle m . In addition, five equality constraints were enforced. The first three constraints required that the steady-state activations produce steady-state muscle forces (f_m^*) that would, if applied, give rise to accelerations \ddot{q}^* equal to the desired accelerations \ddot{q}_d :

$$\ddot{q}^* - \ddot{q}_d = 0. \quad (9)$$

A steady-state muscle force was defined as the force that would result from activation a_m^* after contraction dynamics have equilibrated, i.e., when the fiber velocity is equal to the overall musculotendon velocity projected along the fiber direction ($\dot{l}_m^* = \dot{l}_m^* \cos \alpha_m$). More specifically, f_m^* was computed by scaling the active force–length–velocity (f_{lv}) surface by a_m^* , adding the passive force (f_{passive}), and multiplying by the cosine of the pennation angle:

$$f_m^* = \{a_m^* \cdot f_{lv}(l_m^*, \dot{l}_m^*) + f_{\text{passive}}(l_m^*)\} \cdot \cos(\alpha_m^*), \quad (10)$$

where, for muscle m , α_m^* is the pennation angle at the steady-state muscle length l_m^* . The steady-state muscle length was found using a root solver to balance the force

between each muscle and its tendon (Delp and Loan, 2000). Two additional constraints were used to track the radial pedal forces:

$$f_r^* - f_{r,\text{exp}} = 0, \quad (11)$$

$$f_l^* - f_{l,\text{exp}} = 0, \quad (12)$$

where f_r^* and f_l^* are the right and left steady-state pedal forces generated in the model in response to the steady-state activations (\vec{a}^*), and $f_{r,\text{exp}}$ and $f_{l,\text{exp}}$ are the right and left experimental pedal forces. Tangential pedal forces were not tracked because they are directly related to the crank accelerations, which were tracked. Thus, the static optimization problem was to find values of the activations \vec{a}^* that minimized Eq. (8) and satisfied Eqs. (9), (11), and (12).

Feedback gains $k_p = 400$, $k_v = 40$, and $k_u = 10$ were used. Without actuator dynamics, these gains would achieve critically damped error dynamics with a time constant of 50 ms (Eq. (5)).

The musculoskeletal model and the code describing activation and contraction dynamics were produced using SIMM and the Dynamics Pipeline (Delp and Loan, 2000). Equations of motion for the pedaling model were derived using SD/FAST (Parametric Technology Corporation, Waltham, MA). Optimizations were performed using a sequential quadratic programming algorithm (FSQP; Lawrence et al., 1997; AEM Design, Tucker, GA).

3. Results

The computed muscle control algorithm generated a forward dynamic simulation of pedaling that closely tracked experimental data (Fig. 3) using approximately 10 min of processor time on a 1.7 GHz Pentium IV. The simulated pedal and crank angles were within one standard deviation of experimental measurements throughout the entire pedaling cycle (RMS errors were $\leq 1^\circ$). Simulated pedal forces were within one standard deviation of experimentally measured pedal forces for a majority of the crank cycle. RMS errors were 17 N for the tangential pedal force and 37 N for the radial pedal force.

The time intervals during which the muscles were excited were in relatively close agreement with intervals of excitation estimated from electromyographic recordings (Fig. 4). The timings of the excitations also were similar to the timings of the pulse excitations obtained when simulated annealing was used to compute muscle excitation patterns (Neptune and Hull, 1998). However, the computed muscle control solution produced continuously variable excitation patterns in contrast to the pulse excitation pattern generated with simulated annealing.

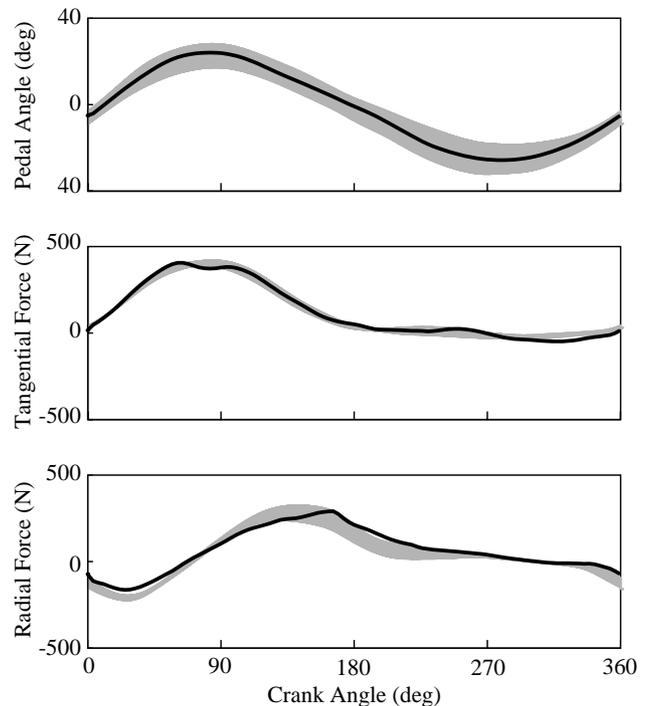


Fig. 3. Simulated (solid black line) and experimental (shaded curves) pedal angles and pedal forces. Experimental data represents the mean (plus and minus one standard deviation) experimental values recorded from ten cyclists pedaling at 60 RPM (Neptune et al., 1997).

4. Discussion

Our objective was to develop a faster method for generating muscle-actuated simulations of movement that track experimental data. Computed muscle control achieves improved computational performance over dynamic optimization by using state feedback and static optimization to compute muscle excitations. Because feedback is used, only a single complete integration of the state equations is required, in contrast to the many thousands of integrations required by standard dynamic optimization techniques. A simulated annealing approach, for example, can require between 5000 and 10,000 complete integrations of the state equations to converge to a solution for pedaling (Neptune, 1999). Since each complete integration of the state equations takes about 1 min, simulated annealing would take about 80–160 h to find a solution. By comparison, the computed muscle control algorithm took only 10 min.

The speed of computed muscle control lessens the need to reduce the number of muscles in a model and make simplifying assumptions about muscle excitation patterns. The large computational cost of dynamic optimization, on the other hand, has driven researchers to adopt different strategies to reduce the number of control variables. For example, Raasch et al. (1997) grouped muscles thought to have similar function,

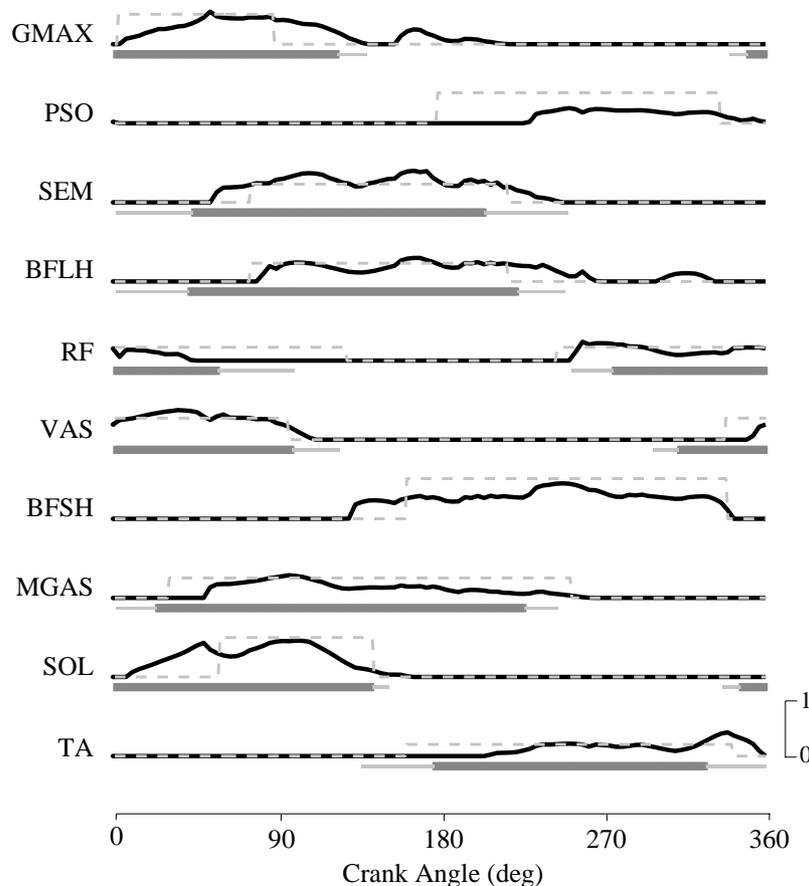


Fig. 4. Estimated muscle excitations (solid black lines) for eleven of the muscles included in the model. The muscles shown are gluteus maximus (GMAX), psoas (PSO), semimembranosus (SEM), biceps femoris long head (BFLH), rectus femoris (RF), vasti (VAS), biceps femoris short head (BFSH), medial gastrocnemius (MGAS), soleus (SOL), and tibialis anterior (TA). For comparison, a simulated annealing solution (dashed lines) and the timing of EMG activity (shaded bars) are given. The simulated annealing solution was obtained using the approach outlined in Neptune and Hull (1998). EMG data represents the mean and standard deviations of muscle onset and offset times for ten cyclists pedaling at 60 RPM (Neptune et al., 1997).

Neptune et al. (1997) assumed pulse excitation patterns, and Pandey et al. (1992) used linear interpolation between a small number of control points. The computed muscle control algorithm, by contrast, generated continuously varying muscle excitations for 30 separate muscles.

The pseudo-inverse method described by Yamaguchi et al. (1995) is fast and allows for continuously varying muscle controls, but it has two important limitations. First, it does not account for activation and contraction dynamics. As a result, unrealistic instantaneous changes in muscle forces can occur, particularly when tracking rapid movements. Second, the pseudo-inverse method assumes that the performance criterion is to minimize the sum of squared muscle stresses. While there is evidence to support the use of this criterion during walking (Crowninshield and Brand, 1981), it may not be ideal for other movements.

In contrast to the pseudo-inverse method, computed muscle control incorporates activation and contraction dynamics in the forward dynamic simulation (Fig. 1, Stage 4). Thus, non-physiologic, instantaneous changes

in muscle force are not possible. In addition, computed muscle control affords flexibility in the formulation of the optimization problem. Performance criteria such as minimum fatigue (Dul et al., 1984), minimum muscle activation (Kaufman et al., 1991), or other criteria can be posed. Experimental data such as measured forces and EMG patterns can also be used to guide the solution by incorporating penalty functions in the performance criterion or by enforcing constraints (e.g., Eqs. (11) and (12)).

Several limitations of using computed muscle control should be recognized. First, although muscle activation and contraction dynamics are included in the forward dynamics stage of the algorithm (Fig. 1, Stage 4), latencies associated with these processes introduce errors between measured and simulated states. At any given time, the muscle forces necessary to produce a desired set of accelerations can be computed, but there are delays between muscle excitations and the production of muscle forces. The effects of these delays were seen where the simulated radial pedal forces lagged slightly behind the experimental values (Fig. 3, crank

angle of $\sim 180^\circ$). Thus, when tracking rapid movements in which actuator delays have larger effects, a method for incorporating actuator dynamics into the estimation of controls may need to be implemented (Lewis et al., 1993). Second, computed muscle control is not a dynamic optimization method; only performance criteria that can be evaluated at an instant in time, such as the sum of squared muscles stresses, can be used. Computed muscle control cannot therefore be used to optimize global measures of performance, such as metabolic energy expended over the duration of a movement. Finally, it should be noted that the ability to use computed muscle control relies on having kinematic data as input and that the quality of the forward dynamic simulation likely depends on the quality of these data.

Inverse dynamics with static optimization (Crowninshield and Brand, 1981), dynamic optimization (Davy and Audu, 1987), and computed muscle control are three techniques that make estimates of muscle forces. Anderson and Pandy (2001) reported that static optimization and dynamic optimization can yield similar estimates of muscle forces for normal gait, and there is no reason to expect that computed muscle control would make substantially different estimates. For more rapid movements, in which activation and contraction dynamics play a large role, differences between static optimization and dynamic optimization solutions might be greater. In these cases, the forces estimated by computed muscle control are influenced by a static optimization performance criterion, but the force trajectories are constrained by activation and contraction dynamics. The ability of computed muscle control to predict muscle forces for different movements is an area for future investigation.

The primary value of computed muscle control is its ability to produce coordinated forward dynamic simulations of movements with relatively little computational expense. These forward dynamic simulations can be used as an initial guess for a dynamic optimization problem, as the basis of perturbation studies (e.g., altering muscle excitations and computing the resulting motions), or as means of conducting detailed investigations of muscle function (e.g., assessing storage and utilization of elastic strain energy). These are capabilities not readily afforded by an inverse dynamics and static optimization approach. Computed muscle control combines the computational speed and practicality of static optimization with the added benefit of producing a well coordinated forward dynamic simulation.

Acknowledgements

We gratefully acknowledge Rick Neptune for comments on the manuscript and for providing the pedaling

model and experimental pedaling data, Jill Higginson for providing a simulated annealing solution for pedaling, Behzad Dariush for discussions about computed torque control, and the support provided to D.G. Thelen by Honda Fundamental Research Labs. Funding for this project was provided by NSF (BES-9702275) and NIH (R01 HD38962, R01 HD33929).

References

- Anderson, F.C., Pandy, M.G., 1999. A dynamic optimization solution for vertical jumping in three dimensions. *Computer Methods in Biomechanics and Biomedical Engineering* 2, 201–231.
- Anderson, F.C., Pandy, M.G., 2001. Static and dynamic optimization solutions for gait are practically equivalent. *Journal of Biomechanics* 34, 153–161.
- Crowninshield, R.D., Brand, R.A., 1981. A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics* 14, 793–801.
- Davy, D.T., Audu, M.L., 1987. A dynamic optimization technique for predicting muscle forces in the swing phase of gait. *Journal of Biomechanics* 20, 187–201.
- Delp, S.L., Loan, J.P., 2000. A computational framework for simulating and analyzing human and animal movement. *IEEE Computing in Science and Engineering* 2, 46–55.
- Delp, S.L., Loan, J.P., Hoy, M.G., Zajac, F.E., Topp, E.L., Rosen, J.M., 1990. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions of Biomedical Engineering* 37, 757–767.
- Dul, J., Johnson, G.E., Shiavi, R., Townsend, M.A., 1984. Muscular Synergism-II. A minimum fatigue criterion for load sharing between synergistic muscles. *Journal of Biomechanics* 17, 675–684.
- Fregly, B.J., 1993. The significance of crank load dynamics to steady-state pedaling biomechanics: an experimental and computer modeling study. Ph.D. thesis, Stanford University, Stanford, CA.
- Fregly, B.J., Zajac, F.E., 1996. A state-space analysis of mechanical energy generation, absorption, and transfer during pedaling. *Journal of Biomechanics* 29, 81–90.
- Kaplan, M.L., Heegaard, J.H., 2001. Predictive algorithms for neuromuscular control of human locomotion. *Journal of Biomechanics* 34, 1077–1083.
- Kaufman, K.R., An, K.N., Litchy, W.J., Chao, E.Y.S., 1991. Physiological prediction of muscle forces-I: theoretical formulation. *Journal of Biomechanics* 40, 781–792.
- Lawrence, C.T., Jzhou, J.L., Tits, A.L., 1997. User's Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints. Technical Report TR-94-16r1. University of Maryland, College Park.
- Lewis, F.L., Abdallah, C.T., Dawson, D.M., 1993. *Control of Robot Manipulators*. Macmillan Publishing Company, New York.
- Neptune, R.R., 1999. Optimization algorithm performance in determining optimal controls in human movement analyses. *Journal of Biomechanical Engineering* 121, 249–252.
- Neptune, R.R., Hull, M.L., 1998. Evaluation of performance criteria for simulation of submaximal steady-state cycling using a forward dynamic model. *Journal of Biomechanical Engineering* 120, 334–341.
- Neptune, R.R., Kautz, S.A., Hull, M.L., 1997. The effect of pedaling rate on coordination in cycling. *Journal of Biomechanics* 10, 1051–1058.
- Pandy, M.G., Anderson, F.C., Hull, D.G., 1992. A parameter optimization approach for the optimal control of large-scale

- musculoskeletal systems. *Journal of Biomechanical Engineering* 114, 450–460.
- Piazza, S.J., Delp, S.L., 2001. Three-dimensional dynamic simulation of total knee replacement motion during a step-up task. *Journal of Biomechanical Engineering* 123, 599–606.
- Raasch, C.C., Zajac, F.E., Ma, B., Levine, W.S., 1997. Muscle coordination of maximum-speed pedaling. *Journal of Biomechanics* 30, 595–602.
- Riley, P.O., Kerrigan, D.C., 1998. Torque action of two-joint muscles in the swing period of stiff-legged gait: a forward dynamic model analysis. *Journal of Biomechanics* 31, 835–840.
- Schutte, L.M., Rodgers, M.M., Zajac, F.E., Glaser, R.M., 1993. Improving the efficacy of electrical stimulation-induced leg cycle ergometry: an analysis based on a dynamic musculoskeletal model. *IEEE Transaction on Rehabilitation Engineering* 1, 109–125.
- Yamaguchi, G.T., Moran, D.W., Si, J., 1995. A computationally efficient method for solving the redundant problem in biomechanics. *Journal of Biomechanics* 28, 999–1005.
- Yamaguchi, G.T., Zajac, F.E., 1990. Restoring unassisted natural gait to paraplegics via functional neuromuscular stimulation: a computer simulation. *IEEE Transactions of Biomedical Engineering* 37, 886–902.
- Zajac, F.E., 1989. Muscle and tendon: properties, models, scaling and application to biomechanics and motor control. *CRC Critical Reviews in Biomedical Engineering* 17, 359–411.
- Zajac, F.E., 1993. Muscle coordination of movement: a perspective. *Journal of Biomechanics* 26, 109–124.