7º CONGRESSO NACIONAL DE BIOMECÂNICA
Paulo Flores *et al.* (Eds)
Guimarães, Portugal, 10 e 11 de fevereiro de 2017

A CO-INTEGRATION APPROACH FOR THE FORWARD-DYNAMICS BASED SOLUTION OF THE MUSCLE RECRUITMENT PROBLEM

Florian Michaud [1], Urbano Lugrís [2] and Javier Cuadrado [3]

[1] *Laboratorio de Ingeniería Mecánica, Universidade da Coruña, Spain; florian.michaud@udc.es*

[2] *Laboratorio de Ingeniería Mecánica, Universidade da Coruña, Spain; ulugris@udc.es*

[3] *Laboratorio de Ingeniería Mecánica, Universidade da Coruña, Spain; javicuad@cdf.udc.es*

KEYWORDS: Muscle force estimation, Forward dynamics, Human motion

ABSTRACT: *The forward-dynamics approach to the estimation of muscle forces during human motion has the best potential to provide insight into muscle coordination. Currently, the computed muscle control (CMC) is the established method in the biomechanics community to implement the mentioned approach. In this work, an alternative method is proposed based on a co-integration scheme, in which the integration of the equations of motion leads the process and the muscular activation and contraction dynamic equations are solved with a smaller time-step size. The accuracy and efficiency of the proposed method are compared with those of the CMC.*

1 INTRODUCTION

It is well-known that using a forward-dynamics approach to find the excitation patterns that best generate movement trajectories has the best potential to provide insight into muscle coordination [1]. Even though the forward and inverse-dynamics models are identical, the estimated muscle forces or optimal excitations by inverse-dynamics approaches hardly reproduce the measured movement in a forward-dynamics simulation due to both errors from the estimation of intersegmental moments by inverse dynamics and errors from numerical integration.

Nowadays, the computed muscle control (CMC) [2] is the established method in the biomechanics community to implement the forward-dynamics approach for muscle force estimation. Basically, the CMC method runs a physiological static optimization [3] to obtain an estimate of muscle excitation, considered constant along the current time step, and then, with such an input, integrates, in a unified scheme, the model equations of motion and the muscle activation and contraction dynamic equations.

In this work, an alternative method is proposed. The idea is similar to the co-integration approach adopted in [4] for a simulation of multibody and hydraulic dynamics: the integration of the equations of motion leads the process and the muscular activation and contraction dynamic equations are solved with a smaller time-step size as possessing higher frequencies.

Since both CMC and co-simulation methods are based on principles that are also used in the simpler physiological static method, the latter is also described in this paper.

2 MULTIBODY FORMULATION

The skeletal system is modeled as a rigid multibody system using independent coordinates. The general equations of such a system, for a vector of generalized coordinates **q**, is:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{Q}_m + \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}}) \quad (1)$$

where **M** is the mass matrix, **H** is the centrifugal and Coriolis forces vector, $\mathbf{Q}_m$ represents the joint torques produced by the muscle forces, and **Q** contains all the remaining applied generalized forces, such as gravity, springs, dampers, etc.

The joint torques $\mathbf{Q}_m$ are produced by muscle contraction forces **F**, which are related to the corresponding joint torques by means of the position-dependent matrix of moment arms $\mathbf{J}_m(\mathbf{q})$:

$$\mathbf{Q}_m = \mathbf{J}_m(\mathbf{q})\mathbf{F} \quad (2)$$

The CNS controls the muscular forces by means of the neural excitations **u**, which will be considered as an external input from the musculoskeletal system's point of view. The following Section explains how the muscular forces are related to the neural inputs and the skeletal motion.

3 MUSCULOTENDON MODEL

Muscles are modeled using the well-known Hill's model. As shown in Figure 1, a musculotendon element is considered as a passive nonlinear elastic tendon (*SE*) in series with a muscle, which is itself formed by two elements acting in parallel: a passive nonlinear elastic spring (*PE*), playing the role of the elasticity of the muscle fibers, and an active contractile element (*CE*). The contractile element is driven by the muscle activation *a*, which varies between 0 and 1.

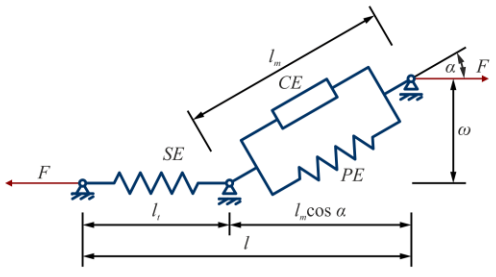

Fig. 1 Hill's muscle model

The force produced by the contractile element F_{CE} is essentially proportional to the activation *a* and the maximum isometric force F^0. However, it is also affected by the muscle elongation and contraction velocity, an effect that can be represented by two nonlinear functions f_l and f_v:

$$F_{CE} = aF^0 f_l(l_m) f_v(\dot{l}_m) \quad (3)$$

Muscles are considered to have a constant width ω, so their fiber pennation angle α will depend on the elongation. Taking this into account, the constitutive equations of *SE*, *PE* and *CE* can be combined into a single nonlinear first-order ODE of the following form [3]:

$$\dot{F} = f_F\left(a, F, l, \dot{l}\right) \quad (4)$$

In this equation, *l* corresponds to the total length of the musculotendon element, which is in turn completely determined by the position of the skeletal system **q**.

The muscle activation *a* is itself governed by another first-order differential equation:

$$\dot{a} = f_a\left(a, u\right) \quad (5)$$

in which the excitation *u* is considered as an independent input, ranging as well between 0 and 1. Excitation-activation dynamics can be understood as a first-order low pass filter: *a* follows *u* with a certain delay, and high-frequency variations of *u* are filtered out. The activation and deactivation time constants of this first-order system lie in general within the 10-50 millisecond range, leading to a significant electromechanical delay.

4 PHYSIOLOGICAL STATIC METHOD

Static optimization methods aim at finding a history of optimal muscle forces that, for a given captured motion, produce joint torques equivalent to those obtained from inverse-dynamics analysis:

$$\mathbf{Q}_m = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{Q}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) \quad (6)$$

Traditionally, static optimization methods consist of stating, at every time step *n*, an optimization problem of the form:

$$\min_{\mathbf{F}^n} f(\mathbf{F}^n) \quad \text{s.t.} \begin{cases} \mathbf{J}_m^n \mathbf{F}^n = \mathbf{Q}_m^n \\ 0 \le \mathbf{F}^n \le \mathbf{F}^0 \end{cases} \tag{7}$$

However, this does not take into account the additional limitations introduced by musculotendon dynamics. Firstly, the smoothing behavior of excitation-activation dynamics does not allow abrupt force variations between consecutive time steps. And secondly, the length and contraction velocity of the active element, which have a relevant effect on the maximum force, cannot be directly related to the skeletal motion, due to the additional degree of freedom introduced by the tendon.

The physiological static method, instead of just imposing constant bounds to muscle forces, aims at taking advantage of the muscular dynamics to get physiologically feasible force limits. Given a time step n, where the muscle forces and activations are known, the idea is to find the minimum and maximum forces a muscle could produce at time step $n+1$, and use them as boundaries for the optimization problem. This is achieved by integrating the equations of musculotendon dynamics with the excitation set to 0 to find the minimum force:

$$\begin{Bmatrix} a_{min}^{n+1} \\ F_{min}^{n+1} \end{Bmatrix} = \begin{Bmatrix} a^n \\ F^n \end{Bmatrix} + \int_{t^n}^{t^{n+1}} \begin{Bmatrix} f_a(a, u=0) \\ f_F(a, F, l, \dot{l}) \end{Bmatrix} dt \tag{8}$$

and to 1 for the maximum force:

$$\begin{Bmatrix} a_{max}^{n+1} \\ F_{max}^{n+1} \end{Bmatrix} = \begin{Bmatrix} a^n \\ F^n \end{Bmatrix} + \int_{t^n}^{t^{n+1}} \begin{Bmatrix} f_a(a, u=1) \\ f_F(a, F, l, \dot{l}) \end{Bmatrix} dt \tag{9}$$

In order to consider the coupling with the skeletal motion, the musculotendon lengths l and elongation velocities $\dot{l}$, which are known at the beginning and end of the interval, are interpolated in time by means of a cubic polynomial.

Then, a constrained optimization problem can be stated for the N muscles at t^{n+1}:

$$\min_{\mathbf{F}^{n+1}} \sum_{i=1}^{N} \left(\frac{F_i^{n+1}}{F_{i,max}^{n+1}} \right)^2 \quad \text{s.t.} \begin{cases} \mathbf{J}_m^{n+1} \mathbf{F}^{n+1} = \mathbf{Q}_m^{n+1} \\ \mathbf{F}_{min}^{n+1} \le \mathbf{F}^{n+1} \le \mathbf{F}_{max}^{n+1} \end{cases} \tag{10}$$

In this case, the chosen objective function is the sum of muscle forces, normalized by their respective maximum values and squared.

After the optimization problem has converged, a root solver is used to calculate the excitations $\mathbf{u}^n$ which, if held constant from t^n to t^{n+1}, would yield the optimum forces $\mathbf{F}_{opt}^{n+1}$ at the end of the interval. For each muscle, the root solver seeks for the value of u that fulfills the lower part of the following expression:

$$\begin{Bmatrix} a_{opt}^{n+1} \\ F_{opt}^{n+1} \end{Bmatrix} = \begin{Bmatrix} a^n \\ F^n \end{Bmatrix} + \int_{t^n}^{t^{n+1}} \begin{Bmatrix} f_a(a, u) \\ f_F(a, F, l, \dot{l}) \end{Bmatrix} dt \tag{11}$$

At the first time step, the process needs initial values for the muscle activations and forces. In order to get force limits for the first optimization, the muscle equations are also integrated with u=0 and u=1, but this time keeping l and $\dot{l}$ constant, and integrating until muscle forces stabilize at steady-state values. This provides absolute maximum and minimum force limits for the current state of the skeletal system. After obtaining the limits, the same optimization and root solver procedure can be used to get the optimum forces and their corresponding activations for the initial step.

Figure 2 represents a flowchart of the algorithm. The dark grey blocks represent time integration of the muscle equations: in the "F limits" block, two integrations per muscle are carried out at every time step, and in the "Root Solver" block, a variable number of muscle integrations is performed until convergence is achieved.

5 COMPUTED MUSCLE CONTROL

The Computed Muscle Control method (CMC) was first introduced by Thelen et al.

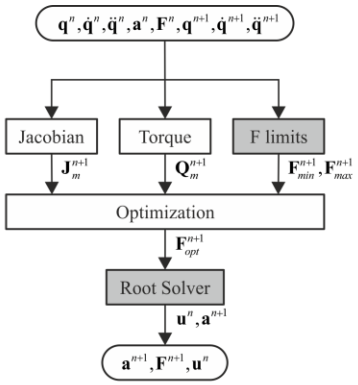

Fig. 2 Static optimization algorithm flowchart.

in 2003, and an updated version of the algorithm was presented later in 2006 [2]. The method calculates muscle excitations at discrete intervals, following a procedure similar to the physiological static method, and then integrates the whole system of equations one step forward, in a unified scheme, using the previously obtained excitations as inputs.

The procedure for calculating the excitations differs from that used in the static method in two aspects: on the one hand, the calculation of the required joint torques $\mathbf{Q}_m$ is performed differently, since the forward integration requires a feedback controller for stabilization. On the other hand, in forward dynamics the positions and velocities at the next time step are unknown, so now their values are estimated using information from the desired motion.

For obtaining the joint torques, a set of controller accelerations is first calculated:

$$\ddot{\mathbf{q}}_{CMC}^{n+1} = \ddot{\mathbf{q}}_d^{n+1} + k_v\dot{\boldsymbol{\varepsilon}}^n + k_p\boldsymbol{\varepsilon}^n \qquad (12)$$

where $\boldsymbol{\varepsilon}^n$ and $\dot{\boldsymbol{\varepsilon}}^n$ represent the position and velocity errors at t^n, k_p and k_v are their corresponding feedback gains, and the subindex d denotes magnitudes related to the *desired* (i.e. tracked) motion. The position and velocity errors are defined as:

$$\begin{aligned}\boldsymbol{\varepsilon}^n &= \mathbf{q}_d^n - \mathbf{q}^n \\ \dot{\boldsymbol{\varepsilon}}^n &= \dot{\mathbf{q}}_d^n - \dot{\mathbf{q}}^n\end{aligned} \qquad (13)$$

If the velocity gain k_v is set as $2\sqrt{k}_p$ and the accelerations $\ddot{\mathbf{q}}_{CMC}^{n+1}$ are assumed to be

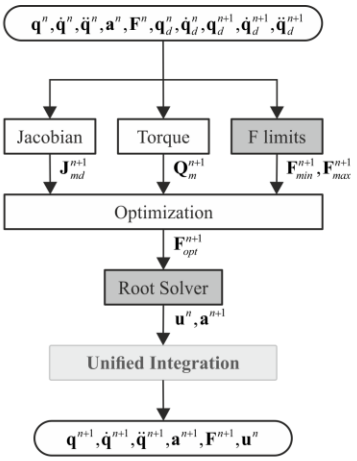

Fig. 3 CMC algorithm flowchart.

reached, the position error would converge to zero in a critically-damped manner [2].

The joint torques that would produce the $\ddot{\mathbf{q}}_{CMC}^{n+1}$ accelerations at the next time step can be now estimated as:

$$\mathbf{Q}_m^{n+1} = \mathbf{M}_d^{n+1}\ddot{\mathbf{q}}_{CMC}^{n+1} - \mathbf{H}_d^{n+1} + \mathbf{Q}_d^{n+1} \qquad (14)$$

And then, the same procedure employed in the static method for calculating the optimal excitations is used here (calculate force limits, optimize forces, and solve for excitations) but, as noted above, using the desired positions and velocities whenever data from the next time step is required.

After the excitations have been obtained, the whole system of differential equations (multibody and musculotendon dynamics) is integrated together from instant n to $n+1$ at a smaller time step, using the calculated excitations as inputs. Figure 3 shows the flowchart of the algorithm, where the similarities and differences with the static method can be clearly observed.

6 CO-SIMULATION METHOD

The CMC method is very fast and robust, but it requires the unified integration of the multibody and muscle dynamics equations. By using a co-integration scheme, the multibody equations can be generated and integrated by an already existing multibody code, while the muscle dynamics are simulated within a different framework.

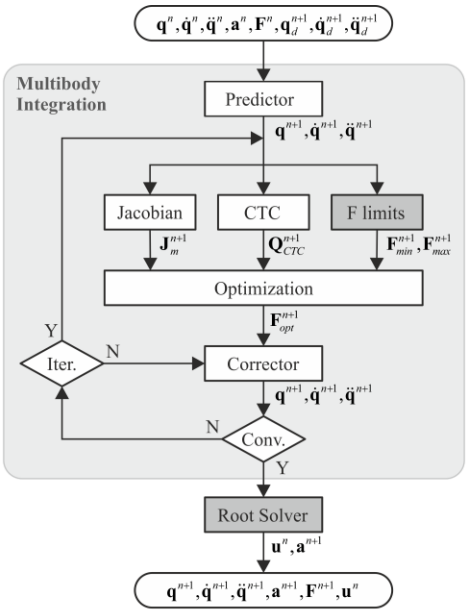

Fig. 4 Co-simulation algorithm flowchart.

The multibody equations are integrated using an implicit integrator, in a predictor-corrector scheme. Therefore, the state estimates at step $n+1$ improve after every iteration, and can be used within the corrector loop for performing new muscle optimizations. The required joint torques for the muscle optimizations are calculated at every iteration using a CTC control algorithm:

$$\begin{aligned} \ddot{\mathbf{q}}_{CTC}^{n+1} &= \ddot{\mathbf{q}}_{d}^{n+1} + k_v \dot{\boldsymbol{\varepsilon}}^{n+1} + k_p \boldsymbol{\varepsilon}^{n+1} \\ \mathbf{Q}_{m}^{n+1} &= \mathbf{M}^{n+1} \ddot{\mathbf{q}}_{CTC}^{n+1} - \mathbf{H}^{n+1} + \mathbf{Q}^{n+1} \end{aligned} \tag{15}$$

After a muscle optimization has converged, the torques $\mathbf{J}_m\mathbf{F}_{opt}$ produced by the optimal forces are introduced in the multibody equations, at the corrector stage. When the corrector loop converges, a root solver is used as before to obtain the excitations.

This method is computationally more expensive than CMC, since several muscle optimizations will be potentially carried out at every time step. However, results are not so different if the optimization is performed only once at the first iteration, using the predictor estimate. This defines a simplified version of the algorithm: Figure 4 shows the flowchart, where the leftmost block represents the choice between performing optimization at every iteration or not.

7 EXAMPLE AND RESULTS

To test the proposed method and compare it with CMC, the simple pendulum actuated by two muscles shown in Figure 5 was used. The pendulum has a massless bar with a tip mass, and its position is defined by the angle with respect to the vertical θ, which is zero when the pendulum is in equilibrium.

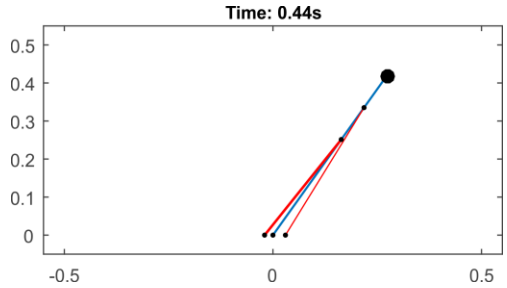

Fig. 5 Simple pendulum actuated by two muscles.

To play the role of the experimentally acquired motion in biomechanical examples, a predefined history of the angle θ was imposed. The pendulum starts from an inverted position (pointing upwards), i.e. $\theta=\pi$, and performs a one-second-long movement described by a continuous quintic spline through the values shown in Table 1.

Tab. 1 Pendulum trajectory

T (s)	θ (deg)
0	180
1/3	180–60
2/3	180+60
1	180

The prescribed motion has zero velocity and acceleration at both $t=0$ and $t=1$. The motion is performed with the pendulum in an inverted position, in order to better test the robustness of the controllers.

The tests were all programmed in Matlab. In order to accelerate the simulations, the muscle equations and their derivatives were implemented in a vectorized MEX-file. All the time integrations of muscle equations related to the optimization (represented by dark grey blocks in Figures 2, 3 and 4) have

been carried out using the *ode23t* integrator, which is an implicit trapezoidal rule.

In the co-simulation method, the multibody system was integrated with a 10 ms time step, using an implicit trapezoidal rule with Newton-Raphson iteration.

The CMC method also used the trapezoidal rule, although integrating the multibody and muscle dynamics equations together, and using a time step of 1 ms. The optimization process was repeated every 10 ms.

Tab. 2 Simulation results

Method	CPU time (s)	RMS error (deg)
Static	7.70	-
CMC	8.25	0.0025
CSF	23.66	0.0186
CSS	7.85	0.1146

Table 2 shows the CPU time and tracking accuracy obtained from the different methods: Static, CMC, CoSimulation Full (CSF), and CoSimulation Simplified (CSS). All the methods performing an optimization every 10 ms required similar computational efforts. However, the simple model analyzed in this paper is not totally conclusive, since the multibody equations (in this case a single equation) are too simple and have almost no computational impact.

CMC achieved the best accuracy due to the smaller integration time step. Co-simulation can reach even better accuracy at 1 ms, but at the cost of a much higher CPU-time due to the large number of optimizations.

Figure 6 (top) shows the tracking errors, and the bottom plot compares the excitations obtained with CMC and CSF. CMC excitations are stepped because the integration is performed at 1 ms, but the excitations are calculated every 10 ms. It can be seen that the results at coinciding time steps are very close. The results from the remaining methods are not shown due to the small differences: if the muscles can always deliver the required torques from Eq.(15), all methods yield very similar excitations.

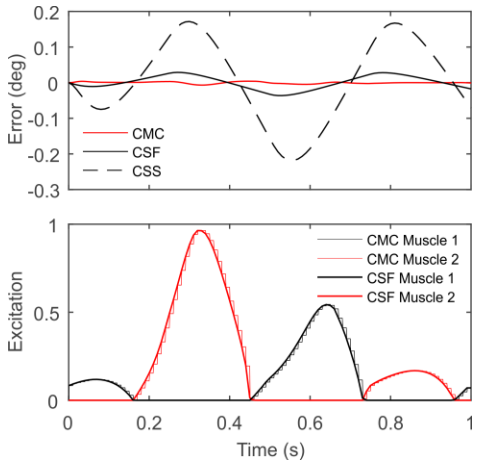

Fig. 6 Results comparison.

In case the motion is more violent and the muscles cannot accurately follow the desired motion, the differences between methods increase, due to the different estimations they use for future positions and velocities. This should never happen when analyzing a recorded motion: if the motion actually happened, muscles were capable of producing it. However, this problem can be interesting in motion prediction.

ACKNOWLEDGMENTS

This work was funded by the Spanish MINECO under project DPI2015-65959-C3-1-R, cofinanced by the EU through the EFRD program.

REFERENCES

[1] F.E. Zajac, "Muscle coordination of movement: a perspective", Journal of Biomechanics, Vol. 35, 1011 – 1018, 1993.

[2] D.G. Thelen, F.C. Anderson, "Using computed muscle control to generate forward dynamic simulations of human walking from experimental data", Journal of Biomechanics, Vol. 39, 1107 – 1115, 2006.

[3] Y. Ou, "An Analysis of Optimzation Mehtods for Identifying Muscle Forces in Human Gait", University of Duisburg-Essen, Duisburg, 2012.

[4] M.A. Naya, J. Cuadrado, D. Dopico, U. Lugris, "An efficient unified method for the combined simulation of multibody and hydraulic dynamics: comparison with simplified and c-integration approaches", The Archive of Mechanical Engineering, Vol. 58, 223 – 243, 2011.