**Direct methods for predicting movement biomechanics based upon optimal control theory with implementation in OpenSim**

**Porsa, Sina, Lin, Yi-Chung and Pandy, Marcus**

# DIRECT METHODS FOR PREDICTING MOVEMENT BIOMECHANICS BASED UPON OPTIMAL CONTROL THEORY WITH IMPLEMENTATION IN OPENSIM

Sina Porsa, Yi-Chung Lin, and Marcus G. Pandy

Department of Mechanical Engineering, University of Melbourne, Parkville, Victoria 3010, Australia

**REVISION 2**

Annals of Biomedical Engineering

9 December 2015

Word count (Abstract through References): 5,987

Running header: Predictive optimal control simulations of movement

Corresponding author:

Marcus G. Pandy, Ph.D.

Department of Mechanical Engineering

University of Melbourne

Parkville, Victoria 3010

Australia

## ABSTRACT

The aim of this study was to compare the computational performances of two direct methods for solving large-scale, nonlinear, optimal control problems in human movement. Direct shooting and direct collocation were implemented on an 8-segment, 48-muscle model of the body (24 muscles on each side) to compute the optimal control solution for maximum-height jumping. Both algorithms were executed on a freely-available musculoskeletal modeling platform called OpenSim. Direct collocation converged to essentially the same optimal solution up to 249 times faster than direct shooting when the same initial guess was assumed (3.4 hours of CPU time for direct collocation versus 35.3 days for direct shooting). The model predictions were in good agreement with the time histories of joint angles, ground reaction forces and muscle activation patterns measured for subjects jumping to their maximum achievable heights. Both methods converged to essentially the same solution when started from the same initial guess, but computation time was sensitive to the initial guess assumed. Direct collocation demonstrates exceptional computational performance and is well suited to performing predictive simulations of movement using large-scale musculoskeletal models.


**Keywords:** direct shooting, collocation, musculoskeletal model, motion tracking, trajectory optimization, predictive simulation.

## INTRODUCTION

Inverse- and forward-dynamics methods are commonly used in conjunction with optimization theory to calculate muscle forces during movement[22,31]. Inverse dynamics uses measured joint motion and ground force data as inputs to a model to determine the net moments applied about each joint. Static optimization is then applied to solve the net moment – muscle force redundancy problem[3,9]. Because this approach solves a different optimization problem at each time instant, the goal of the motor task (e.g., minimizing metabolic energy over one gait cycle) is not easily incorporated in the formulation of the problem[22].

Forward-dynamics or optimal control theory uses neural excitations as inputs to drive a model of the neuromusculoskeletal system in a forward simulation of movement. Optimal control theory presents the most powerful framework for determining muscle forces because this approach incorporates a model of both the system dynamics and the goal of the motor task (i.e., the performance criterion) in the formulation of the optimization problem[3,4,10,13,23,25]. Computed muscle control[28] and neuromuscular tracking[26] are two recent approaches that use optimal control theory to track experimental gait data. This formulation of the optimal control problem is often referred to as 'state estimation[6,7]. Because the time histories of joint motions and external forces are used explicitly in the calculation of muscle forces, novel movements cannot be predicted using this approach. A more powerful application of optimal control theory is the formulation of a 'trajectory optimization problem', where all quantities of interest (i.e., joint motion, ground forces and muscle activation patterns) are predicted concurrently.

Numerical methods for solving optimal control problems are often divided into two categories: indirect methods and direct methods[6]. An indirect method applies the calculus of variations to derive analytical expressions for the necessary conditions for optimal control, which are specified in terms of the adjoint differential equations, Pontryagin's maximum principle, and the associated boundary conditions[13,25]. The result is a two-point-boundary-value problem that can be challenging to solve for two reasons: firstly, finding an initial guess for the adjoint variables is problematic as these quantities often have no physical interpretation; and secondly, backward integration of the adjoint differential equations is numerically unstable because these equations are highly nonlinear.

Direct methods do not require either analytical expressions for the necessary conditions for optimal control or an initial guess for the adjoint variables. Instead, the control and/or state variables are approximated using parameterization and the optimal control problem is transcribed to a nonlinear programming problem. Direct shooting is the numerical method used to solve the optimal control problem when only the control variables are parameterized[3,4,8,14,19,23], whereas direct collocation involves parameterization of both the control and state variables[1,17,27].

Whilst the application of direct shooting results in fewer design variables compared to direct collocation, direct shooting is computationally more expensive as small time steps are needed to integrate a set of stiff differential equations during the optimization procedure. Furthermore, a gradient-based optimizer would require additional CPU time to calculate the derivatives of the performance criterion and constraints with respect to the controls because each perturbation of the controls requires a numerical integration of the system dynamical equations[13,22]. Anderson and Pandy[3] computed a minimum-metabolic-energy solution for walking with the body represented as a 3D, 23-degree-of-freedom skeleton actuated by 54 muscles. A direct shooting method[23] required the model equations to be integrated ~900 times per iteration of the computational algorithm. Convergence to the optimal solution took ~10,000 hours of CPU time using multiple processors on an IBM SP-2 parallel supercomputer, making this method impractical.

Direct collocation offers a more efficient means of solving large-scale, nonlinear, optimal control problems as both the states and controls are discretized and the system dynamical equations are converted into algebraic constraints, thus circumventing the need for explicit integration. In addition, the nonlinear programming problem is characterized by a sparsely populated constraint Jacobian matrix that can be solved efficiently. Direct collocation has been combined with musculoskeletal modeling to solve state estimation (tracking) problems and produce stable forward simulations of human movement[17, 29]. However, relatively few studies have used this approach to solve a trajectory optimization problem and predict movement biomechanics independent of experimental data[1,2,12,18,20,27]. Stelzer and von Stryk[27] and Eriksson[12] applied simple models of the body to simulate kicking and arm lifting, respectively. Ackermann and van den Bogert[1] used a 9-degree-of-freedom skeleton actuated by 16 muscles to study how changes in the performance criterion affect the optimal control solution computed for normal walking. Kaplan and Heegard[17] solved a tracking problem with a simplified model of human pedaling dynamics and showed that direct collocation converges more quickly than direct shooting. To our knowledge, no study has quantitatively compared the computational performance of direct collocation to that of direct shooting when these two methods are used to solve the same trajectory optimization problem for human movement. In addition, no study has described the sensitivity of large-scale optimal control solutions to a change in the initial guess.

The overall goal of the present study was to compare the computational performances of the direct shooting and direct collocation methods in solving large-scale optimal control problems in human movement. Both methods were implemented on a detailed musculoskeletal model of the body to compute the optimal control solution for maximum-height jumping. This task was chosen because it presents a relatively unambiguous performance criterion (i.e., to maximize the height reached by the center of mass (COM) of the body). Our specific aims were to (1) implement the direct shooting and direct collocation computational algorithms on a freely-available musculoskeletal modeling platform called OpenSim; (2) compare the CPU times needed to converge to the optimal solutions computed by direct shooting and direct collocation using the same initial guess; (3) quantitatively evaluate the optimal solution derived from each method against the time histories of joint angular displacements, ground reaction forces, and muscle EMG activity measured for subjects jumping to their maximum achievable heights; and (4) determine the sensitivity of the optimal control solution to a change in the initial guess.

## METHODS

### *Musculoskeletal model*

A model of the body was created based on the 'Generic Gait2392 model' available in OpenSim (version 3.2)[11]. The skeleton was modeled as a planar, 8-segment, 10-degree-of-freedom linkage (Fig. 1). The pelvis was connected to the ground via a 3-degree-of-freedom planar joint with two translations and one rotation. The head, arms and torso were represented by a single rigid body that articulated with the pelvis via a hinge joint. The hip and ankle were each modeled as single-degree-of-freedom hinge joints whereas the knee was represented as a 1-degree-of-freedom translating hinge joint[11]. Joint hyperextension was prevented by applying a passive torque at each joint that represented the actions of the ligaments. The skeleton was actuated by 48 muscle-tendon units (24 muscles on each side of the body), with each unit represented as a Hill-type muscle in series with an elastic tendon. Muscle lines-of-action were represented in 3D based on the attachment sites assumed in the generic OpenSim model.

Eight spheres were placed under the model foot to simulate the interaction of the foot with the ground; five spheres were located under the hind-foot and three were placed under the toes (Fig. 1). The normal contact force applied by each sphere was found by summing the Hertzian contact force[16] and a Hunt-Crossley damping force[15]. The stiffness and damping coefficients of the contact spheres were found by solving optimization problems that reproduced the vertical ground forces measured for walking at the self-selected speed (1.4 m/s) and for running at 7 m/s (Supplementary Material). These tasks were chosen because they capture all possible movements of the foot (i.e., making, maintaining and breaking contact with the ground) across a wide range of locomotion speeds. The fore-aft component of the ground force was assumed to arise purely from friction, and a value of 0.8 was assumed for the static and dynamic friction coefficients[21]. Data defining the structure and parameters of the model are available at https://simtk.org/home/vertical_jump/.

*Optimal control problem*

Jumping was assumed to be bilaterally symmetric[4,24], and so 7 independent generalized coordinates in the model (i.e., anterior-posterior and vertical pelvic translations, pelvic tilt, lumbar extension, and flexion-extension of the hip, knee and ankle) were actuated by 24 independent muscle excitations. The system dynamical equations can be represented explicitly as:

$$\begin{aligned}
\underline{\dot{q}} &= \frac{d\underline{q}}{dt} \\
\underline{\ddot{q}} &= -M^{-1}(\underline{q})\left(\underline{C}(\underline{q},\underline{\dot{q}}) + \underline{G}(\underline{q}) + R(\underline{q})\,\underline{F}^{MT}(\underline{q},\underline{l}^{M}) + \underline{T}_{foot}(\underline{q},\underline{\dot{q}}) + \underline{T}_{lig}(\underline{q},\underline{\dot{q}})\right), \\
\underline{\dot{l}}^{M} &= h_1(\underline{q},\underline{l}^{M},\underline{a}); \quad 0 \le \underline{a} \le 1 \\
\underline{\dot{a}} &= h_2(\underline{a},\underline{u})
\end{aligned} \quad (1)$$

where $\underline{q}$ is a 7x1 vector of generalized coordinates; $\underline{\dot{q}}$ is a 7x1 vector of generalized velocities; $\underline{l}^{M}$ is a 24x1 vector of muscle-fiber lengths; $\underline{a}$ is a 24x1 vector of muscle activations; $\underline{u}$ is a 24x1 vector of muscle excitations (controls); $M(\underline{q})$ is a 7x7 mass matrix; $\underline{C}(\underline{q},\underline{\dot{q}})$ is a 7x1 vector of centrifugal and Coriolis forces and torques; $\underline{G}(\underline{q})$ is a 7x1 vector of gravitational forces and torques; $R(q)$ is a 7x24 matrix of muscle moment arms; $\underline{F}^{MT}$ is a 24x1 vector of muscle-tendon forces; $\underline{T}_{foot}(\underline{q},\underline{\dot{q}})$ is a 7x1 vector of spring forces used to simulate foot-ground interaction; and $\underline{T}_{lig}(\underline{q},\underline{\dot{q}})$ is a 7x1 vector of ligament and damping torques. Equation (1) can be expressed in compact form as:

$$\underline{\dot{x}} = \underline{f}(\underline{x},\underline{u},t)\,. \quad (2)$$

where $\underline{x}$ is a 62x1 state vector defined by $\underline{x} = (\underline{q},\underline{\dot{q}},\underline{l}^{M},\underline{a})^{T}$. For prescribed values of the controls and states at each time instant, equation (2) was evaluated using OpenSim API (version 3.2) by executing a MEX-function that provided an interface between OpenSim and MATLAB.

The performance criterion was to maximize the height reached by the COM during the jump[25] :

$$J = Y_C(t_f) + \frac{\dot{Y}_C^2(t_f)}{2g} \quad (3)$$

where $Y_C(t_f)$ and $\dot{Y}_C(t_f)$ are the vertical position and velocity of the COM at the final lift-off time, $t_f$; and $g$ is the gravitational acceleration constant. An inequality constraint was imposed to prevent the COM from countermoving prior to upward propulsion:[5]

$$\ddot{Y}_C(t_n) < \ddot{Y}_C(t_{n+1}); \quad n = 1,2,...,7\,, \tag{4}$$

where $\ddot{Y}_C(t_n)$ is the vertical acceleration of the COM at the $n^{th}$ temporal node. Equation (4) was enforced to reproduce the conditions under which human subjects performed a non-countermovement jump from an initial squatting position. A set of inequality constraints was also imposed to constrain the values of the muscle excitations:

$$0 \le \underline{u}(t) \le 1, \tag{5}$$

where $u = 1$ and $u = 0$ represent the fully excited and fully de-excited states of a muscle, respectively. Additional upper and lower bounds were imposed on the state variables when implementing the direct collocation method:

$$\underline{x}_{\min} \le \underline{x}(t) \le \underline{x}_{\max}\,. \tag{6}$$

Thus, the optimal control problem was to maximize the performance criterion (equation (3)) subject to the system dynamical equations (equation (2)) and a set of linear inequality constraints (equations (4)-(5) for direct shooting and equations (4)-(6) for direct collocation).

*Creating an initial guess*

Optimal control solutions for jumping were computed using two different initial guesses. In each case, the model began from a prescribed static squatting position. The first initial guess was created by using computed muscle control[28] to calculate the muscle excitation histories needed to drive the model in a forward simulation to track the joint angular displacements and velocities calculated from a previously published simulation of jumping[4]. Direct shooting and direct collocation were then applied separately to compute the optimal control solution assuming a free final lift-off time, $t_f$ . Direct shooting required an initial guess only for the controls (muscle excitations), whereas direct collocation required initial guesses for both the states and controls.

A second initial guess was created using a previously reported 'bang-bang' optimal control solution for jumping[24] . The bang-bang (on-off) muscle excitation histories were input to the model and the system dynamical equations were integrated to compute the corresponding states. Direct shooting and direct collocation were then applied separately to solve the same free-final-time problem using the second initial guess. The solutions computed for the first and second initial guesses were compared to determine the sensitivity of each method to a change in the initial guess.

*Computation of the optimal controls*

Direct shooting was implemented by discretizing the control histories corresponding to the initial guess and transcribing the optimal control problem to a nonlinear programming problem[14,23]. The time interval $[t_0, t_f]$ was discretized evenly into 28 subintervals and the control histories were parameterized using 29 nodal points. This number of nodes was based on that used to solve the direct collocation problem (see below). Since the model began from a prescribed static position, the values of the control nodes at $t = t_0$ were known. Thus, the optimal control problem for direct shooting was defined by 28*24=672 control variables, a free final time $t_f$ , 7 path constraints (equation (4)), and (672+1)*2=1,346 inequality constraints (equation (5)) (see Table 1).

The direct collocation method was implemented by discretizing both the control and state histories using 29 nodal points and replacing the system dynamical equations with 62*28=1,736 algebraic equality constraints or defects, Δ, using Euler's method:

$$\Delta = (\underline{x}_{n+1} - \underline{x}_n) - (t_{n+1} - t_n)\,.\; f\left(\left(\frac{\underline{x}_n + \underline{x}_{n+1}}{2}\right), \left(\frac{\underline{u}_n + \underline{u}_{n+1}}{2}\right), \left(\frac{t_n + t_{n+1}}{2}\right)\right),\; n = 1,2,...,28. \qquad (7)$$

Twenty-nine nodes were used to reproduce the nodal density selected by Ackerman and van den Bogert[1] in their simulation of human gait (i.e., 1 node per 10 ms of simulation time). Analogous to direct shooting, the values of the controls and states at $t = t_0$ were known. Thus, the optimal control problem for direct collocation was defined by 2,409 design variables (24*28 control variables, 62*28 state variables, and the final time, $t_f$), 7 path constraints (equation (4)), (24*28+62*28+1)*2=4,818 inequality constraints (equations (5)-(6)), and 28*62=1,736 equality constraints (equation (7)) (Table 1).

Both optimal control problems were solved using an interior-point algorithm available within the fmincon function defined in the optimization toolbox of MATLAB (Toolbox Version 7.0 R2014a, MathWorks Inc., Natick, MA). For the direct shooting problem, the derivatives of the performance criterion and constraints with respect to the design variables were calculated using central differences; thus, the system equations were integrated 2*673=1,346 times per iteration of the computational algorithm**.** The optimal control solution was computed until the value of jump height improved by less than $10^{-6}$ m and the values of the constraints were satisfied to a tolerance of $10^{-3}$.

*Efficient computation of the Jacobian matrix*

The defect constraints (equation (7)) were functions of only the two neighboring temporal nodes, and so the constraint Jacobian matrix, which contained the derivatives of the constraints with respect to the states and controls, was sparse (Fig. 2). The computational performance of the direct collocation method can be improved by accounting for this sparse structure. For the jumping problem, the constraint Jacobian comprised of 62*86*28*28+62*28 or 4,182,024 total entries (Fig. 2). However, the vast majority of these entries were zeros; indeed, at most only 2*27*(62*86)+62*86+28*62=294,996 or 7.1% of the entries within the entire grid structure of the Jacobian were non-zero (Fig. 2, blue squares and blue line in bottom panel), leaving 3,887,028 entries or 92.9% of the entire grid structure with zeros. These zero entries were taken into account in computing the optimal control solution using direct collocation. Additional zeros within each block of the constraint Jacobian (Fig. 2, top panel) could not be taken into account because of the built-in functionality provided by OpenSim (see Discussion below). The constraint Jacobian matrix was calculated using central differences to approximate the derivatives (Supplementary Material).

All calculations were performed on a 3.4 GHz PC (Intel® Core™ i7-3770 Processor). Jump height was calculated as the difference between the maximum height reached by the COM during the jump and the height of the COM at standing. The direct shooting and direct collocation solutions were compared against experimental data reported for healthy young subjects jumping to their maximum achievable heights[4].

# RESULTS

*Computational performance*

Direct shooting required 602 iterations and 846.4 hours (35.3 days) of CPU time to converge to the optimal control solution for jumping using the first initial guess (Fig. 3 and Table 2). By comparison, direct collocation required 702 iterations and 3.4 hours of CPU time (249 times faster than direct shooting) to converge to essentially the same optimal solution from the same initial guess. The computational performance of direct collocation remained far superior to that of direct shooting when the second initial guess was used, but the speed-up in CPU time (103 times) was less (9.6 hours versus 990.1 hours (41.3 days)).

*Comparison of model and experiment*

The model simulations reproduced the salient features of maximum-height jumping. The optimal muscle excitation histories predicted by the model were consistent with measured EMG (Fig. 4). In agreement with experiment, the erector spinae, gluteus maximus, vasti, soleus and gastrocnemius were fully activated in the model for the majority of ground contact time. Some differences between model and experiment were also evident. For example, the biarticular hamstring was activated only at the beginning of the simulated jump, whereas subjects activated their hamstrings during the entire propulsion phase (Fig. 4, SEMIMEM).

Peak vertical ground forces predicted by direct shooting and direct collocation were 3.0 BW and 3.1 BW, respectively, compared with measured values ranging from 2.5 BW to 3.0 BW (Fig. 5). Consistent with experiment, the model generated much smaller forces in the fore-aft direction. However, peak fore-aft ground forces predicted by both methods (~0.7 BW) were higher than the range of values recorded from experiment (0.2-0.4 BW).

Trajectories of the vertical position and velocity of the COM predicted by the model were consistent with experimental results (Fig. 5). Direct shooting and direct collocation both predicted a vertical COM velocity at lift-off of 2.2 m/s compared to 2.0-2.5 m/s measured for subjects. Both methods predicted a maximum jump height of ~36 cm, which agreed closely with the values obtained from experiment (mean, 36.9 cm; range: 33-41 cm)[4] (Table 2). Ground contact times predicted by direct shooting and direct collocation were 0.31 sec and 0.28 sec, respectively, which also compared well with the average time of 0.30 sec recorded for subjects (range: 0.25-0.39 sec).

The joint angular displacements predicted by both methods were generally consistent with experiment (Fig. 5). The model extended the hip, knee and ankle joints in a similar fashion to that measured for subjects, although one noticeable difference was that the hip and knee in the model were slightly more flexed during initial propulsion (Fig. 5). The pattern of back extension calculated in the model was substantially different from that recorded from experiment, but there was considerable variability in the experimental data.

*Sensitivity of the optimal solution to the initial guess*

Both methods were sensitive to the initial guess assumed. Although the second initial guess was much farther from the optimal solution than the first, as evidenced by the uncoordinated motion of the joints (Figs 6-7, gray and black stick figures) and a calculated jump height of 0.2 cm lower than the standing height, there were similarities in the optimal joint angles, COM motion, and ground forces computed using the two initial guesses (Figs 6-7, heavy and light lines). However, some clear differences were also evident. For both direct shooting and direct collocation, the peak vertical ground force predicted for the second initial guess was noticeably lower

(~0.5 BW) than that calculated for the first, resulting in a ~7 cm difference in jump height (Table 2).

## DISCUSSION

The aim of this study was to compare the computational performances of two direct methods for solving large-scale, nonlinear, optimal control problems in human movement. Direct shooting and direct collocation were implemented on a detailed musculoskeletal model to compute a free-final-time optimal control solution for maximum-height jumping. Both computational algorithms were executed on an open-source musculoskeletal modeling platform, OpenSim. Direct collocation converged to essentially the same optimal solution up to 249 times faster than direct shooting using the same initial guess. There was good agreement between the optimal joint angles, ground forces, and muscle excitation patterns predicted by the model and corresponding experimental data. Maximum jump heights predicted by the two methods were virtually identical and in close agreement with the mean value obtained from experiment. The model converged to a quantitatively different optimal solution depending on the initial guess assumed. However, both methods converged to the same solution when started from the same initial guess.

Few previous studies have used direct collocation to solve a trajectory optimization problem and predict movement biomechanics independent of experimental data[1,12,27]. The models used by Stelzer and von Stryk[27] and Eriksson[12] were relatively simple with the body represented as a 2-degree-of-freedom linkage and actuated by fewer than 10 muscles. Ackermann and van den Bogert[1] used a planar, 7-segment, 16-muscle model of the body to study how changes in the performance criterion affect the optimal solution computed for gait. In the present study a planar, 8-segment, 48-muscle model of the body (24 muscles on each side) was used to compute the optimal control solution for maximum-height jumping, and the model predictions were quantitatively compared against measurements of the joint angles, COM position and velocity, ground forces, and muscle activation patterns obtained as subjects jumped to their maximum achievable heights. We also compared the computational performances of direct shooting and direct collocation when the same optimal control problem was solved using the same initial guess. In addition, the present study described the sensitivity of large-scale optimal control solutions to a change in the initial guess.

The computational performance of direct collocation was vastly superior to that of direct shooting even though many more iterations were needed to converge to the optimal solution (Table 2). This is because the CPU time per iteration was ~300 times less for direct collocation compared to direct shooting. CPU time per iteration for direct shooting was significantly greater because 2*673=1,346 forward integrations of the system dynamical equations were needed to evaluate the derivatives of the constraints (using central differences) and determine a search direction at each iteration. In contrast, direct collocation solved the system equations implicitly by enforcing a set of defect constraints (equation (7)) at each of the state nodal points. This process involved the simultaneous solution of a set of nonlinear algebraic equations and was less expensive computationally than repeated integration of the system dynamical equations.

The variation in CPU time across iterations was much larger for direct shooting than direct collocation. For direct shooting, CPU times per iteration for the solutions derived from the first and second initial guesses were 84.3 ± 32.4 min and 90.1 ± 38.4 min, respectively, compared to 0.3 ± 0.0 min and 0.3 ± 0.1 min for direct collocation (Table 2). The high standard deviation in CPU time for direct shooting was due to different step sizes used during the forward integrations performed to evaluate the derivatives. When the dynamics of the simulated jump changed rapidly, as evidenced by a rapid change in the vertical ground force, the step size selected by the integrator was necessarily small, causing an increase in the time taken to integrate the system equations. Conversely, when the dynamics of the simulated jump changed more slowly, step size was larger, and the time taken to integrate the system equations was proportionately less. This led to large fluctuations in the CPU time per iteration for direct shooting. In contrast, CPU time per iteration for direct collocation was nearly constant because the time taken to calculate a search direction did not depend on how the dynamics of the simulated jump was changing.

In theory a global optimal control solution can be computed from an arbitrary initial guess. We found that the optimal solutions computed for maximum-height jumping using direct shooting and direct collocation were slightly different (i.e., jump height was 5-7 mm higher for direct shooting), and neither can be guaranteed to represent the global optimum. There are at least two explanations for this result. Firstly, different discretization schemes were used to approximate the continuous-time dynamics for the present direct shooting and direct collocation problems. We used Euler's method with 29 evenly-spaced nodes to solve the collocation problem, whereas an OpenSim built-in 5$^{th}$-order Runge-Kutta method with a variable step size was used to solve the direct shooting problem. Secondly, the derivatives of the performance criterion and constraints were evaluated differently during a solution of the direct shooting and direct collocation problems, and this information was then used by a nonlinear programming solver to determine a search direction. To illustrate the effects of each of these factors on the computation of the optimal solution, we solved a standard optimal control problem using the direct shooting and direct collocation methods (Supplementary Material). This analysis clearly demonstrated that both methods again converged to different solutions regardless of the size of the optimal control problem.

The model converged to a quantitatively different optimal solution depending on the initial guess assumed (Figs 6-7). In particular, optimal jump heights predicted by the two methods using the second initial guess (29.9 cm and 29.4 cm for direct shooting and direct collocation, respectively) were much lower than those calculated using the first initial guess (36.7 cm and 36.0 cm, respectively). This result is most likely explained by the nature of gradient-based algorithms, which may have difficulty converging when the initial guess is at some distance from the neighborhood of the optimal solution. This was demonstrated in the present study where both methods converged to a sub-optimal solution for the second initial guess with a maximum jump height that was 7 cm lower than that computed for the first initial guess. These results emphasize the fact that computation of the optimal controls is sensitive to the initial guess and that caution should be exercised in generating an initial guess. For the direct collocation method, an initial guess is needed for both the control and state trajectories. One option is to apply computed muscle control or neuromuscular tracking to compute the muscle excitations needed to track joint angle trajectories measured from experiment[26,28]. However, this approach is limited when the aim is to predict novel movements that result from changes in a model parameter such as muscle strength. In these instances, the best approach may be to solve the direct collocation problem repeatedly using a series of different initial guesses to gain confidence that the solution indeed represents a global optimum.

A noticeable difference between the two methods was the strategy used to optimize the performance criterion. Jump height increased rapidly during the first few iterations of the direct collocation solution, accompanied by relatively large errors in the constraints (Fig. 8, red lines). As the constraint errors were subsequently reduced, jump height also decreased markedly. Once the constraint errors were sufficiently small, however, jump height increased nearly monotonically prior to convergence. In contrast, the direct shooting solution was characterized by a more gradual improvement in jump height with much smaller constraint errors generated throughout (Fig. 8, blue lines). The constraint errors were larger for the collocation solution because ~5,000 additional constraints (1,736 defect constraints plus 3,472 upper and lower bounds on the state variables) were prescribed for this problem compared with that for direct shooting (Table 1). Because the system equations were integrated explicitly in direct shooting, only a set of linear inequality constraints on the control variables (equation (5)) and 7 non-linear path constraints (equation(4)) had to be satisfied, making it possible for the optimizer to find a search direction that improved jump height without committing large errors in the constraints.

The computational performance of direct collocation remained far superior to that of direct shooting when the second initial guess was used, but the speed-up in CPU time was less; direct collocation computed the optimal solution 103 times faster for the second initial guess compared to 249 times faster for the first initial guess. This result may be explained by the contrasting strategies used by these two methods to compute the optimal control solution for jumping. Figure 8 shows that the constraint errors increased by more than a factor of two during the first few iterations of the direct collocation solution when the second initial guess was used, which required a larger number of iterations for convergence. In contrast, the constraint errors generated during the first few iterations of the direct shooting solution were similar for the first and second initial guesses. This finding suggests that the computational performance of direct collocation is more sensitive to a change in the initial guess than that of direct shooting.

To our knowledge, this is the first study to implement direct collocation in OpenSim, a widely used open-source modeling and simulation environment for studying movement biomechanics. Previous studies have used custom musculoskeletal models and software to solve trajectory optimization problems in human movement[1,3,10,13,20,27,30]. The present study demonstrates the potential of performing predictive simulations of movement using direct collocation within OpenSim. While the results are encouraging, there are some limitations which require further consideration.

First, the structure of the sparsely populated constraint Jacobian matrix was not fully exploited when solving the collocation problem. As described in the Methods section, the constraint Jacobian comprised of 62*86*28*28+62*28=4,182,024 total entries (Fig. 2). By examining an exploded view of a representative block of the Jacobian grid structure (Fig. 2, blue squares in bottom panel), it can be seen that only 406 entries (Fig. 2, blue dots in top panel) out of 62*86=5,332 total entries in each block were non-zero, leaving 4,926 entries or 92.4% of each block containing zeros. Thus, only 2*27*406+406+28*62=24,066 entries or 0.6% of all entries within the constraint Jacobian were non-zero for the jumping problem. The results of Fig. 3 did not account for the zero entries within each block, which meant that 4,926*(2*27)+4,926=270,930 additional zeros (6.5% of all entries in the constraint Jacobian) were unnecessarily computed while forming the derivatives of the constraints. We were unable to account for the additional zeros within each block because calculation of the derivatives of the constraints is currently limited to the built-in functionality provided by OpenSim. Future work should focus on developing the software needed to customize these calculations so that the sparse structure of the constraint Jacobian matrix can be fully exploited.

Second, an explicit formulation of the dynamical equations of motion was used in the present study. An optimal control problem can be solved either by explicitly (i.e., equation (2)) or implicitly formulating the equations for neuromusculoskeletal dynamics. The latter approach has been shown to be more efficient computationally because analytical expressions are given for the various entries of the constraint Jacobian[29]. Recent studies have applied implicit direct collocation and derived optimal control solutions for human movement using less than 30 minutes of CPU time[20,30]. While it is theoretically possible to solve our jumping problem using implicit direct collocation, OpenSim currently provides only an explicit formulation of the dynamical equations of motion, and a significant amount of customization would therefore be needed to export the implicit form of these equations from OpenSim. This issue warrants careful consideration when contemplating the solution of an optimal control problem in OpenSim using direct collocation.

## ACKNOWLEDGMENTS

This work was supported by a VESKI Innovation Fellowship awarded to MGP. A University of Melbourne Postgraduate Scholarship to SP is also gratefully acknowledged.

## REFERENCES

1. Ackermann M. and A. J. van den Bogert. Optimality principles for model-based prediction of human gait. Journal of Biomechanics 43: 1055-1060, 2010.

2. Ackermann M. and A. J. van den Bogert. Predictive simulation of gait at low gravity reveals skipping as the preferred locomotion strategy. Journal of Biomechanics 45: 1293-1298, 2012.

3. Anderson F. C. and M. G. Pandy. Dynamic optimization of human walking. Journal of Biomechanical Engineering 123: 381-390, 2001.

4. Anderson F. C. and M. G. Pandy. A dynamic optimization solution for vertical jumping in three dimensions. Comput Methods Biomech Biomed Engin 2: 201-231, 1999.

5. Anderson F. C. and M. G. Pandy. Storage and utilization of elastic strain energy during jumping. Journal of Biomechanics 26: 1413-1427, 1993.

6. Betts J. T. Survey of numerical methods for trajectory optimization. Journal of Guidance, Control, and Dynamics 21: 193-207, 1998.

7. Bryson A. E. Applied optimal control: Optimization, estimation and control. CRC Press, 1975.

8. Celik H. and S. J. Piazza. Simulation of aperiodic bipedal sprinting. Journal of Biomechanical Engineering 135: 081008-081008, 2013.

9. Crowninshield R. D. Use of optimization techniques to predict muscle forces. Journal of Biomechanical Engineering 100: 88-92, 1978.

10. Davy D. T. and M. L. Audu. A dynamic optimization technique for predicting muscle forces in the swing phase of gait. Journal of Biomechanics 20: 187-201, 1987.

11. Delp S. L., F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman and D. G. Thelen. Opensim: Open-source software to create and analyze dynamic simulations of movement. IEEE Transactions on Biomedical Engineering 54: 1940-1950, 2007.

12. Eriksson A. Optimization in target movement simulations. Computer Methods in Applied Mechanics and Engineering 197: 4207-4215, 2008.

13. Hatze H. The complete optimization of a human motion. Mathematical Biosciences 28: 99-135, 1976.

14. Hull D. G. Conversion of optimal control problems into parameter optimization problems. Journal of Guidance, Control, and Dynamics 20: 57-60, 1997.

15. Hunt K. and F. Crossley. Coefficient of restitution interpreted as damping in vibroimpact. Journal of Applied Mechanics 42: 440-445, 1975.

16. Johnson K. Contact mechanics. Cambridge University Press, 1985.

17. Kaplan M. L. and J. H. Heegaard. Predictive algorithms for neuromuscular control of human locomotion. Journal of Biomechanics 34: 1077-1083, 2001.

18. Kistemaker D. A., J. D. Wong and P. L. Gribble. The cost of moving optimally: Kinematic path selection. Journal of Neurophysiology 112: 1815-1824, 2014.

19. Miller R. H., S. C. Brandon and K. J. Deluzio. Predicting sagittal plane biomechanics that minimize the axial knee joint contact force during walking. Journal of Biomechanical Engineering 135: 011007, 2013.

20. Miller R. H. and J. Hamill. Optimal footfall patterns for cost minimization in running. Journal of Biomechanics 2015.

21. Miller R. H., B. R. Umberger and G. E. Caldwell. Limitations to maximum sprinting speed imposed by muscle mechanical properties. Journal of Biomechanics 45: 1092-1097, 2012.

22. Pandy M. G. Computer modeling and simulation of human movement. Annual Review of Biomedical Engineering 3: 245-273, 2001.

23. Pandy M. G., F. C. Anderson and D. G. Hull. A parameter optimization approach for the optimal control of large-scale musculoskeletal systems. Journal of Biomechanical Engineering 114: 450-460, 1992.

24. Pandy M. G. and F. E. Zajac. Optimal muscular coordination strategies for jumping. Journal of Biomechanics 24: 1-10, 1991.

25. Pandy M. G., F. E. Zajac, E. Sim and W. S. Levine. An optimal control model for maximum-height human jumping. Journal of Biomechanics 23: 1185-1198, 1990.

26. Seth A. and M. G. Pandy. A neuromusculoskeletal tracking method for estimating individual muscle forces in human movement. Journal of Biomechanics 40: 356-366, 2007.

27. Stelzer M. and O. Von Stryk. Efficient forward dynamics simulation and optimization of human body dynamics. ZAMM Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 86: 828-840, 2006.

28. Thelen D. G. and F. C. Anderson. Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. Journal of Biomechanics 39: 1107-1115, 2006.

29. van den Bogert A. J., D. Blana and Heinrich. Implicit methods for efficient musculoskeletal simulation and optimal control. Procedia IUTAM 2: 297-316, 2011.

30. Van den Bogert A. J., M. Hupperets, H. Schlarb and B. Krabbe. Predictive musculoskeletal simulation using optimal control: Effects of added limb mass on energy cost and kinematics of walking and running. Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology 1754337112440644, 2012.

31. Zajac F. E. and M. E. Gordon. Determining muscle's force and action in multi-articular movement. Exercise and Sport Sciences Reviews 17: 187-230, 1989.

## FIGURE CAPTIONS

Fig. 1: Schematic diagram of the musculoskeletal model used in this study. Only those muscles on the right side of the body are shown for clarity. Foot-ground contact was modeled using eight contact spheres per foot (inset). Forces were applied to the model foot at the points of ground contact. There is no contact force between the spheres even though some spheres overlap with each other as illustrated in this figure.

Fig. 2: Structure of the constraint Jacobian matrix computed for the musculoskeletal model used in this study (see Fig. 1). The $i^{th}$ block-row of the 28*28 grid (bottom panel) represents the derivatives of all defect constraints with respect to the control and state variables at the $i^{th}$ node. Because the defect constraints were a function of two neighboring nodes, each row contained only two non-zero blocks (blue squares); the entries of the remaining 26 blocks were all zeros (empty squares). Please note that the derivatives of all defect constraints with respect to the final time variable were all non-zeros and were represented as a blue line at the bottom of the grid. Further improvement in computational speed may be obtained by exploiting the sparse structure of the Jacobian matrix inside each block as illustrated in the exploded view given in the top panel. Each row of each block contained 62 entries related to the defect constraints as defined in equation (7) (i.e., 7 entries related to joint angular displacements ($\Delta_q$);7 entries related to joint angular velocities ($\Delta_{\dot{q}}$); 24 entries related to muscle-fiber lengths ($\Delta_L$); and

24 entries related to muscle activations ($\underline{\Delta}_a$)). Each column of each block contained 86 entries associated with the state and control variables (i.e., 7 joint angular displacements ($\underline{q}$), 7 joint angular velocities ($\underline{\dot{q}}$), 24 muscle-fiber lengths ($\underline{L}$), 24 muscle activations ($\underline{a}$), and 24 controls ($\underline{u}$)). Thus, the size of each block was 62*86=5,332 entries while that of the entire Jacobian matrix was 5332*28*28+62*28 =4,182,024 entries.

Fig. 3: Top: Stick figures showing the configuration of the model at lift-off as a function of iteration number. Iteration number is expressed as a percentage of the total number of iterations performed using direct shooting and direct collocation in conjunction with the first initial guess (left panel) and the second initial guess (right panel).

Bottom: Total CPU time plotted against the absolute number of iterations taken to converge to an optimal control solution for jumping using the direct shooting method (blue line) and the direct collocation method (red line) for the first initial guess (left panel) and the second initial guess (right panel).

Fig. 4: Muscle excitation histories representing the first initial guess (black lines), optimal muscle excitations predicted by direct shooting starting from the first initial guess (blue lines), and optimal muscle excitations predicted by direct collocation starting from the first initial guess (red lines). Also shown are muscle EMG activation patterns recorded for one subject executing a maximum-height jump (gray lines)[4]. EMG data shown are for the subject that most closely matched the BMI of the model. EMG data for each muscle were normalized by the maximum electrode voltage recorded during a maximal voluntary contraction[4]. The vertical axes for the model muscle excitations and the EMG data therefore range from 0 to 1. Muscle abbreviations used

are: ERCSPN, erector spinae; INTOBL, internal obliques; GMAX, medial gluteus maximus; SEMIMEM, semimembranosus; RF, rectus femoris; VASi, vastus intermedius; GAS, medial gastrocnemius; SOL, soleus.

Fig. 5: Time histories of the vertical and fore-aft ground reaction forces and center-of-mass vertical position and velocity (left column) and the angular displacements of the back, hip, knee and ankle joints (right column) generated by the first initial guess (black lines) compared to the optimal solutions predicted by direct shooting (blue lines) and direct collocation (red lines). The gray lines represent experimental data recorded for five subjects jumping to their maximum achievable heights[4]. Positive values represent back extension, hip flexion, knee extension, and ankle dorsiflexion. The stick figures above each column show the configuration of the body during ground contact time. The gray stick figures represent the experimental data averaged for the five subjects while the red stick figures represent the configuration of the model corresponding to the direct collocation solution. The configuration of the model corresponding to the direct shooting solution is indistinguishable from that corresponding to direct collocation and is therefore not shown.

Fig. 6: Time histories of the fore-aft and vertical ground reaction forces and center-of-mass vertical position and velocity generated by the first and second initial guesses (left column) compared to the optimal solutions predicted by direct shooting (middle column) and direct collocation (right column). In all panels, the light lines represent the first initial guess and its corresponding optimal solutions while the heavy lines represent the second initial guess and its

corresponding optimal solutions. The stick figures above each column show the configuration of the model during ground contact time.

Fig. 7: Time histories of the back, hip, knee and ankle joint angular displacements generated by the first and second initial guesses (left column) compared to the optimal solutions predicted by direct shooting (middle column) and direct collocation (right column). In all panels, the light lines represent the first initial guess and its corresponding optimal solutions while the heavy lines represent the second initial guess and its corresponding optimal solutions. The stick figures above each column show the configuration of the model during ground contact time.

Fig. 8: Variation in the values of the performance criterion (jump height) and maximum constraint error plotted against the percentage of total number of iterations for direct shooting (blue lines) and direct collocation (red lines). The light and heavy lines represent the optimal solutions derived using the first and second initial guesses, respectively. The unit of the maximum constraint error represented on the vertical axis in the bottom panel is not given as it varies according to the state possessing the maximum value.

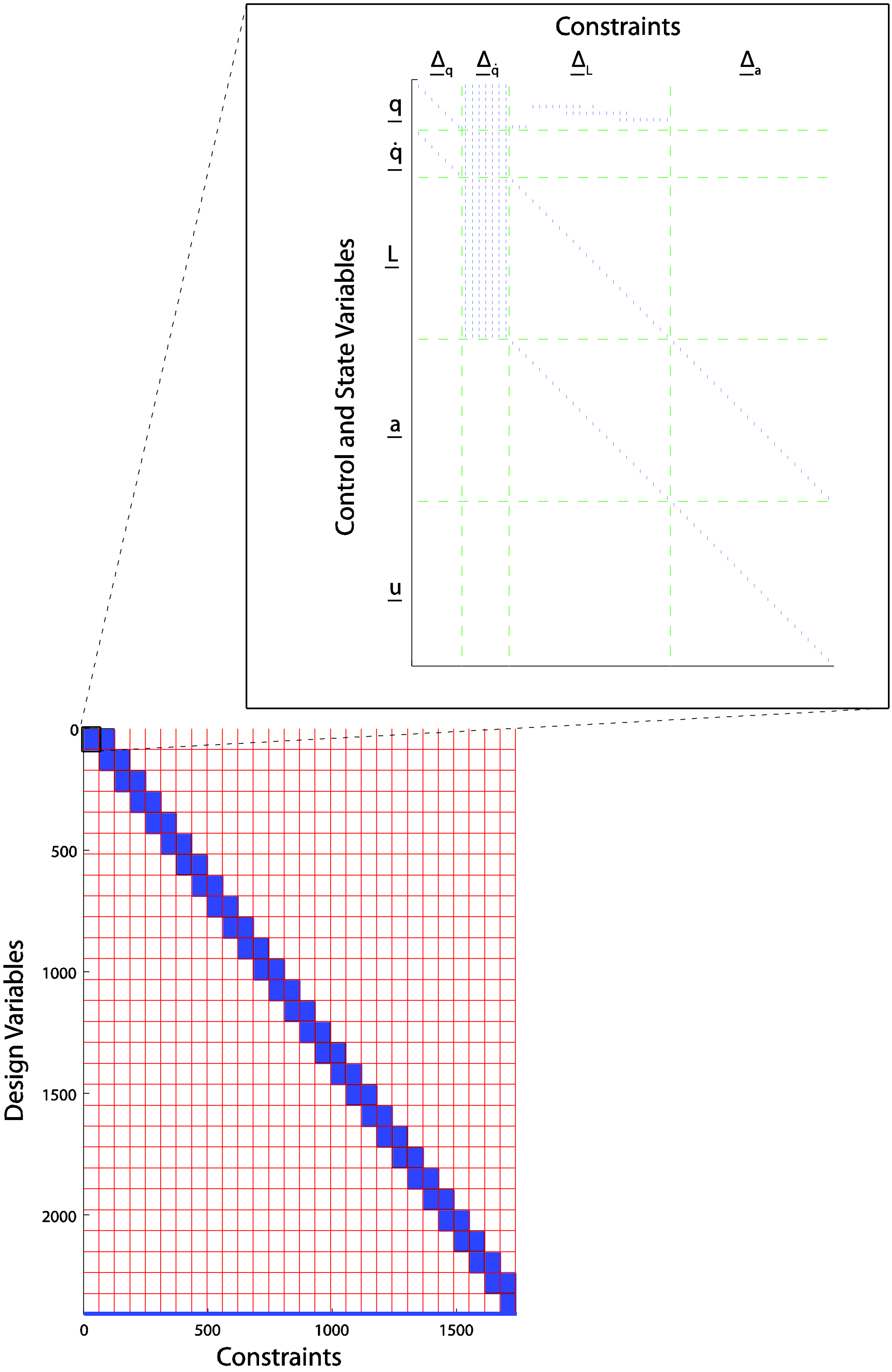
# Tables

Table 1: Comparison of the design variables and constraints used in the formulation of the direct shooting and direct collocation problems. Note that the number of bounds is two times greater than the total number of design variables because an upper and lower bound was applied to each design variable. See text for details.

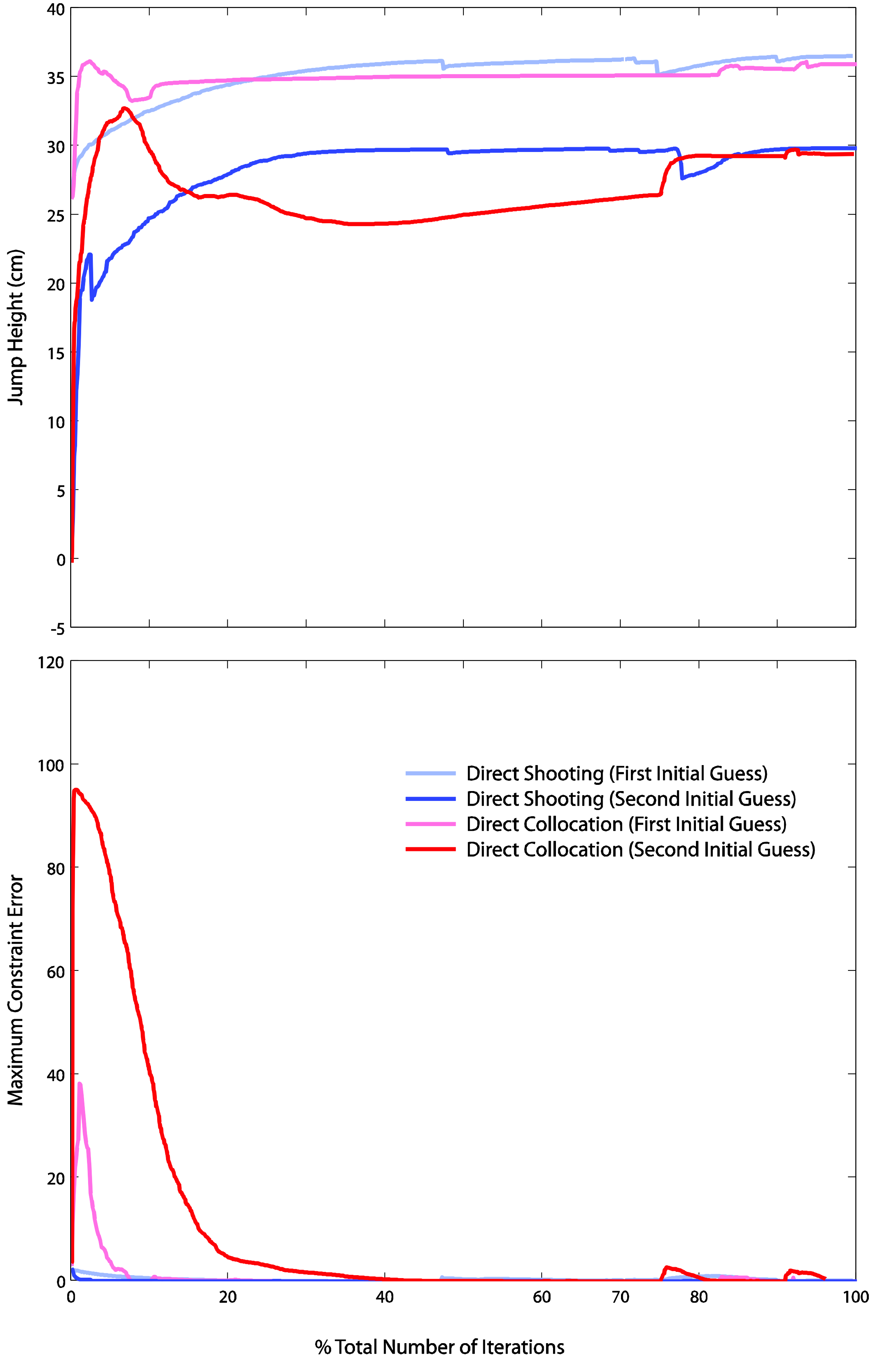
| | Design Variables | | | | Constraints | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Controls | States | Final Time | Total | Path | Defect | Bounds | Total |
| Direct Shooting | 672 | 0 | 1 | 673 | 7 | 0 | 1346 | 1353 |
| Direct Collocation | 672 | 1736 | 1 | 2409 | 7 | 1736 | 4818 | 6561 |

Table 2: Comparison of number of iterations, CPU time, jump height, and ground contact time for the direct shooting and direct collocation optimal control solutions. Average CPU time per iteration is expressed as mean ± one standard deviation.

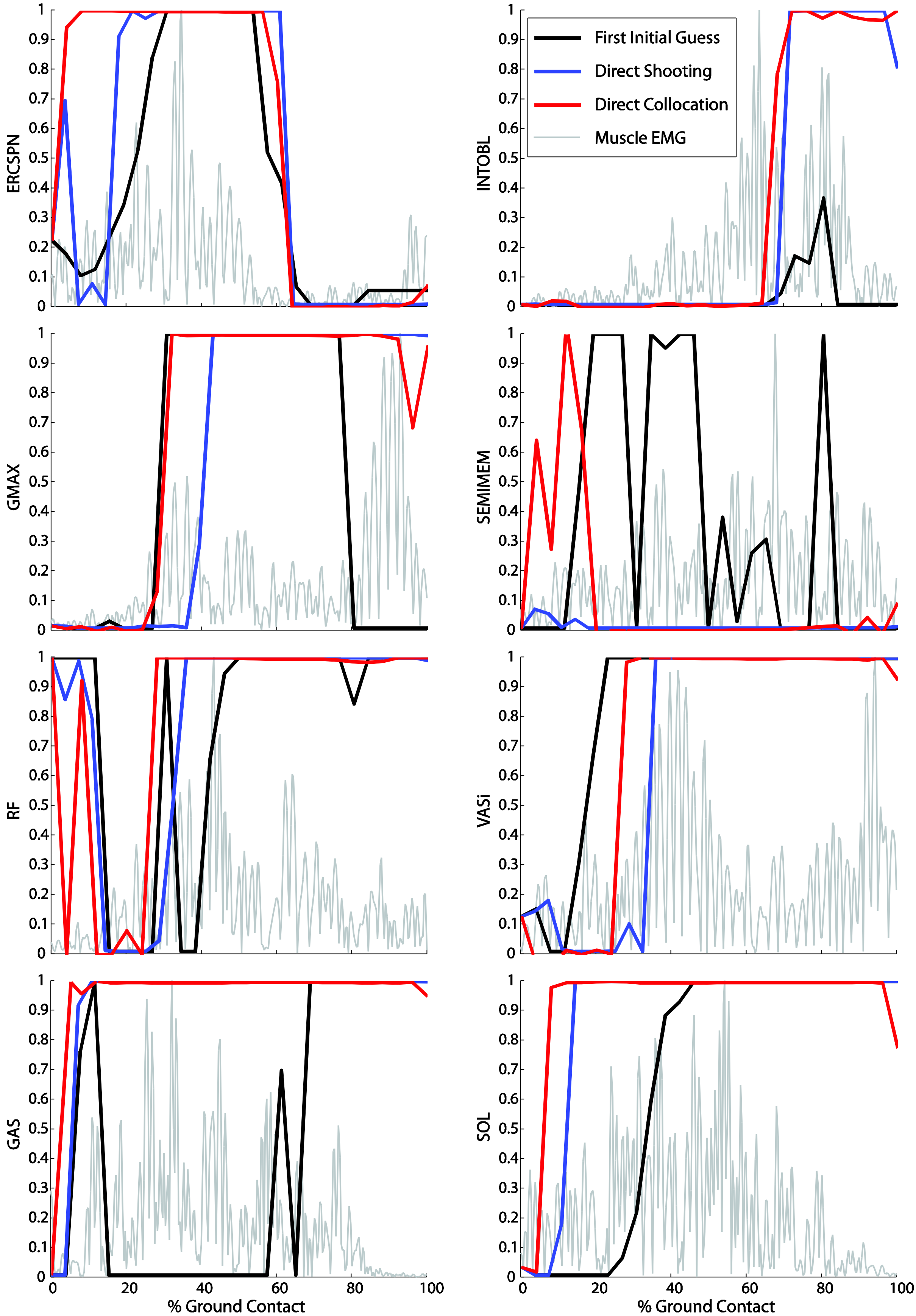
| | Method | Number of Iterations | CPU Time (hours) | CPU Time per Iteration (min) | Jump Height (cm) | Ground Contact Time (sec) |
|---|---|---|---|---|---|---|
| First Initial Guess | Direct Shooting | 602 | 846.4 | 84.3 ± 32.4 | 36.7 | 0.31 |
| | Direct Collocation | 702 | 3.4 | 0.3 ± 0.0 | 36.0 | 0.28 |
| Second Initial Guess | Direct Shooting | 661 | 990.1 | 90.1 ± 38.4 | 29.9 | 0.29 |
| | Direct Collocation | 1683 | 9.6 | 0.3 ± 0.1 | 29.4 | 0.27 |


Constraints
$\underline{\Delta}_q$
$\underline{\Delta}_{\dot{q}}$
$\underline{\Delta}_L$
$\underline{\Delta}_a$
Control and State Variables
$\underline{q}$
$\underline{\dot{q}}$
$\underline{L}$
$\underline{a}$
$\underline{u}$
Design Variables
Constraints
0
500
1000
1500
2000

1


40
35
30
25
20
15
10
5
0
-5
Jump Height (cm)
120
100
80
60
40
20
0
Maximum Constraint Error
Direct Shooting (First Initial Guess)
Direct Shooting (Second Initial Guess)
Direct Collocation (First Initial Guess)
Direct Collocation (Second Initial Guess)
0
20
40
60
70
80
100
% Total Number of Iterations

1


First Initial Guess
Direct Shooting
Direct Collocation
Muscle EMG
ERCSPN
INTOBL
GMAX
SEMIMEM
RF
VASi
GAS
SOL
% Ground Contact
% Ground Contact

Author/s:
Porsa, S;Lin, Y-C;Pandy, MG

Title:
Direct Methods for Predicting Movement Biomechanics Based Upon Optimal Control Theory with Implementation in OpenSim

Date:
2016-08-01

Citation:
Porsa, S., Lin, Y. -C. & Pandy, M. G. (2016). Direct Methods for Predicting Movement Biomechanics Based Upon Optimal Control Theory with Implementation in OpenSim. ANNALS OF BIOMEDICAL ENGINEERING, 44 (8), pp.2542-2557. https://doi.org/10.1007/s10439-015-1538-6.

Persistent Link:
http://hdl.handle.net/11343/283147