Medical Engineering and Physics 54 (2018) 56–64

Contents lists available at ScienceDirect

Medical Engineering and Physics

journal homepage: www.elsevier.com/locate/medengphy

A computational framework for simultaneous estimation of muscle and joint contact forces and body motion using optimization and surrogate modeling

Ilan Eskinazi[a], Benjamin J. Fregly[b,*]

[a] Department of Mechanical & Aerospace Engineering, University of Florida, Gainesville, FL, USA
[b] Department of Mechanical Engineering, Rice University, Houston, TX, USA

ARTICLE INFO

Article history:
Received 24 August 2017
Revised 11 January 2018
Accepted 11 February 2018

Keywords:
Musculoskeletal
Modeling
Contact
Muscle
Optimization
Neural network
Surrogate
Moment arms
Knee
Joint

ABSTRACT

Concurrent estimation of muscle activations, joint contact forces, and joint kinematics by means of gradient-based optimization of musculoskeletal models is hindered by computationally expensive and non-smooth joint contact and muscle wrapping algorithms. We present a framework that simultaneously speeds up computation and removes sources of non-smoothness from muscle force optimizations using a combination of parallelization and surrogate modeling, with special emphasis on a novel method for modeling joint contact as a surrogate model of a static analysis. The approach allows one to efficiently introduce elastic joint contact models within static and dynamic optimizations of human motion. We demonstrate the approach by performing two optimizations, one static and one dynamic, using a pelvis-leg musculoskeletal model undergoing a gait cycle. We observed convergence on the order of seconds for a static optimization time frame and on the order of minutes for an entire dynamic optimization. The presented framework may facilitate model-based efforts to predict how planned surgical or rehabilitation interventions will affect post-treatment joint and muscle function.

1. Introduction

Modeling and simulation of muscle and joint contact forces has the potential to improve patient care for movement-related disorders. Reliable concurrent estimation of these forces along with joint kinematics could be used to predict joint replacement performance, surgical outcomes, and rehabilitation strategies for a variety of musculoskeletal disorders. Most studies that predict muscle and joint contact forces model biological joints as constraint-based engineering joints. In those studies, muscle and joint contact forces are calculated by following a two-step process: (1) Muscle forces are computed using a multibody dynamic skeletal model and optimization, and then (2) Associated joint contact forces are calculated from knowledge of the muscle forces and joint reaction forces from inverse dynamics [1–4]. The downsides of this two-step approach are that it can produce erroneous muscle force predictions [5] and cannot predict secondary kinematics (e.g., knee anterior-posterior translation) or ligament forces for the joints being modeled.

For this reason, researchers have sought to develop more complex modeling methods that allow concurrent estimation of muscle and joint contact forces. Such methods replace constraint-based engineering joints with deformable joint surfaces whose interactions are controlled primarily by muscle and ligament forces. Lin et al. (2010) predicted muscle and knee contact forces simultaneously using a two-level optimization method where the outer level guessed the muscle force distribution and the inner-level found the corresponding static configuration of the joint using surrogate contact models [6]. These models approximated the input-output characteristics of elastic foundation contact models. Thelen et al. (2014) and Smith et al. (2016) used a modified version of computed muscle control (CMC), where a controller tracked desired accelerations while joint translational accelerations were assumed to be zero and an elastic foundation was used to model contact [7,8]. Marra et al. (2015) and Andersen et al. (2011, 2017) used force-dependent kinematics (FDK), where secondary joint coordinates were added as design variables within a static optimization [9–11]. With this approach, the velocities and accelerations of the secondary coordinates were assumed to be zero and an elastic foundation was used to model contact. In one recent study a surrogate contact model was used to speed up FDK computation [12]. Guess et al. (2014)

[*] Corresponding author.
E-mail address: fregly@rice.edu (B.J. Fregly).

https://doi.org/10.1016/j.medengphy.2018.02.002
1350-4533/© 2018 IPEM. Published by Elsevier Ltd. All rights reserved.

avoided optimization and instead used feedback control with deformable contact models for the foot and joint of interest [13]. Moissenet et al. (2014) performed concurrent computation of muscle and contact forces within an optimization by using simplified joint models and Lagrange multipliers [14].

Unfortunately, developing optimization-based predictions of motion where muscle and joint contact forces are solved concurrently remains a difficult and computationally slow task. The primary reasons are the difficulties encountered when applying gradient-based optimization to musculoskeletal models with computationally costly and non-smooth (discontinuous or non-differentiable) contact models as well as non-smooth musculoskeletal geometry models (e.g., muscle-tendon lengths and moment arms). Contact models are computationally costly because they involve computing distances between complex three-dimensional surfaces and are non-smooth when various regions of the contacting surfaces come in to and out of contact. Moreover, contact forces and moments are sensitive to small pose variations that affect normal contact force, resulting in badly scaled gradients when pose parameters defining joint position and orientation are used as design variables [15]. Non-smoothness in muscle-tendon lengths and muscle moment arms can arise when muscles are modeled geometrically using sequences of line segments whose paths are determined by either wrapping objects or via points added or removed as a function of spanned joint angles [16–18]. Non-smoothness can be introduced when a line segment enters contact with a wrapping object, passes through a wrapping surface, snaps to the other side of a wrapping surface, or is rerouted by turning on or off a via point.

In this study, we propose a novel framework for performing concurrent muscle, joint contact, and joint kinematic simulations via optimization. We remove the non-smoothness problem while increasing computational efficiency by: (1) Generating surrogate models of deformable joint contact from finite element models and efficiently implementing them within optimizations using a novel approach, (2) Generating surrogate models of musculoskeletal geometry and using a custom Hill-type muscle-tendon model with rigid tendon, and (3) Parallelizing multibody dynamic model evaluations. The method results in the computationally efficient computation of non-linear constraints that are incorporated into static and dynamic optimizations of muscle and contact forces. In addition to describing the overall approach with special focus on surrogate contact modeling, we provide two illustrative examples to demonstrate implementation of the approach to knee contact and leg muscle force prediction. In the first example, we use a static optimization approach based on the existing FDK framework which we will call modified FDK (mod-FDK). In the second example, we formulate the same problem as a dynamic optimization and solve it using direct collocation.

2. Methods

2.1. Overview

The goal of our framework (Fig. 1) is to remove non-smoothness and computational expense from optimizations that predict muscle forces, joint contact forces, and joint motions simultaneously. We achieve this goal using a combination of surrogate modeling and parallelization. Surrogate modeling generates smooth and computationally inexpensive approximations of more computationally expensive models, while parallelization splits part of the computational load among multiple processors.

2.1.1. Key concepts

Before continuing with the methods, we first introduce several key concepts. The first concept is that of primary and secondary generalized coordinates. Secondary generalized coordinates are assumed to maintain a quasi-static equilibrium between muscle, ligament, and contact forces, disregarding the effect of inertial forces. The time derivatives of the secondary generalized coordinates are always assumed to be zero in our approach. Primary generalized coordinates are assumed to be affected by inertial forces, and thus their time derivatives are not assumed to be zero.

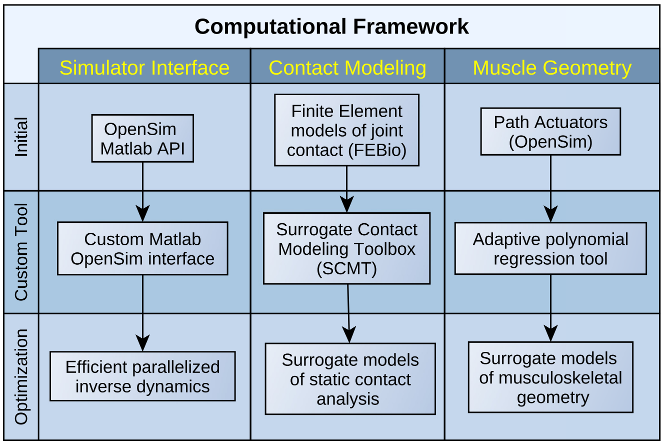
Computational Framework
Simulator Interface
Contact Modeling
Muscle Geometry
Initial
Custom Tool
Optimization
OpenSim Matlab API
Finite Element models of joint contact (FEBio)
Path Actuators (OpenSim)
Custom Matlab OpenSim interface
Surrogate Contact Modeling Toolbox (SCMT)
Adaptive polynomial regression tool
Efficient parallelized inverse dynamics
Surrogate models of static contact analysis
Surrogate models of musculoskeletal geometry

Fig. 1. Summary of computational framework for speeding up and removing non-smoothness from musculoskeletal optimization problems with joint contact. In the initial stage, joint contact is modelled using finite elements, muscle-tendon geometry using path actuators, and the simulator interface is the OpenSim API for Matlab. Three custom tools were developed which allowed us to obtain surrogate models of joint contact and musculoskeletal geometry, as well as an efficient simulation interface for deployment in the optimization stage.

Another important concept is that of static and dynamic optimization. A static optimization performs a minimization at a specified time point, while a dynamic optimization performs a minimization over some period of time. Dynamic optimization may also be known as trajectory optimization or optimal control. While a static optimization minimizes a cost function and is subjected to equality and/or inequality constraints, a dynamic optimizations minimizes a cost functional and is subjected to path constraints and possibly end-point constraints, though other types of constraints can be incorporated as well.

For contact modeling purposes, we also define the concept of a fixed body and a moving body. Since contact forces depend on the relative orientation of one body with respect to another, we define the fixed body as the contacting body that is conceptually fixed while the moving body is thought of as being translated and rotated by six pose parameters (3 translations and 3 rotations) relative to the fixed body.

A final important concept related to contact modeling is that of sensitive directions. A sensitive direction is defined as a degree of freedom (DOF), either translational or rotational, which when perturbed causes relatively large changes in the contact loads (forces and moments resulting from contact) associated with that DOF. The concept of sensitive directions is intimately related to surrogate contact model creation and optimization formulation.

2.1.2. About this framework

In inverse dynamics-based muscle force optimizations, the goal is to find the muscle activations (design variables) that minimize some assumed measure (e.g., fatigue or energy) while the joint forces and moments calculated via inverse dynamics are constrained to be balanced by a combination of muscle, ligament, and contact forces. The focus of our framework is on how to efficiently compute smooth non-linear constraints, representing the balancing of the net joint forces and moments. In the first part of the methods, we explain the general approach required to compute these

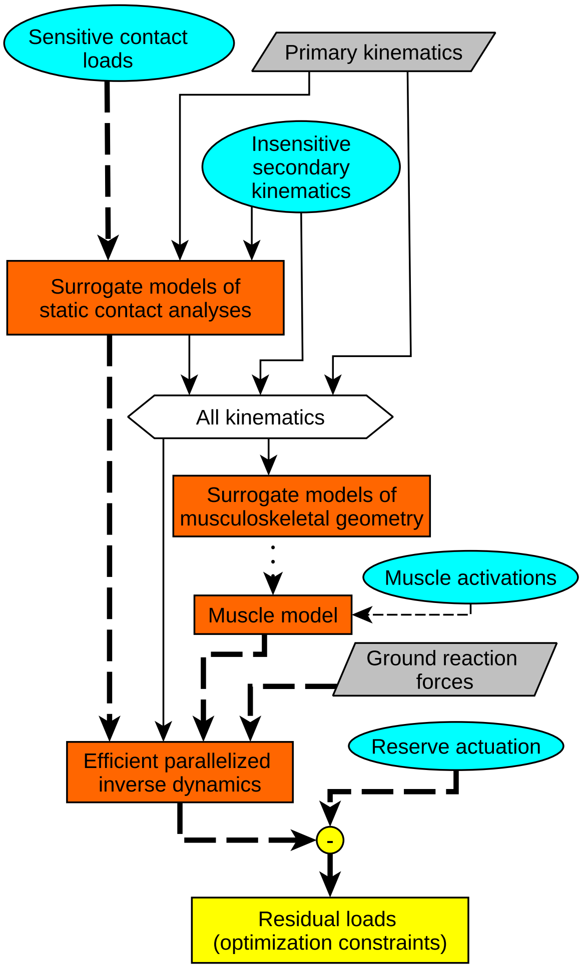
Sensitive contact loads
Primary kinematics
Insensitive secondary kinematics
Surrogate models of static contact analyses
All kinematics
Surrogate models of musculoskeletal geometry
Muscle activations
Muscle model
Ground reaction forces
Efficient parallelized inverse dynamics
Reserve actuation
-
Residual loads (optimization constraints)

Fig. 2. Nonlinear constraint framework for the optimization of muscle activations, joint kinematics, and joint contact loads. Design variables are framed in blue ovals, models are framed in orange rectangles, prescribed data are framed in grey parallelograms, and the set of residual loads is framed in a yellow rectangle. Thick dashed lines represent loads, thin continuous lines represent kinematics, the thin dashed line represents muscle activations, and the thin dotted line represents muscle-tendon lengths, muscle-tendon velocities, and muscle moment arms. Note that the primary kinematics can be also be design variables when the optimization is dynamic. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

constraints. We then explain the implementation of each component involved in evaluating these constraints, which involves (a) formulating and generating surrogate contact models from finite element static analyses, (b) generating surrogate models of musculoskeletal geometry, and (c) computing parallelized skeletal inverse dynamics.

2.2. General approach

Our optimization approach uses the following categories of design variables: (1) Muscle activations, (2) Reserve actuator activations, (3) Contact loads along sensitive directions expressed in the coordinate systems of the "moving" bodies, and (4) Generalized coordinates for secondary kinematics corresponding to insensitive directions (Fig. 2). If the optimization is dynamic, then the generalized coordinates and their first and second time derivatives corresponding to the primary kinematics can be design variables as well, whereas if the optimization is static, the generalized coordinates corresponding to the primary kinematics must be prescribed functions of time. Given design variable categories (3) and (4) and the primary kinematics, surrogate models of static contact analyses output all corresponding loads and poses required to obtain static equilibrium with contact. The generalized speeds and accelerations are assumed to be zero for generalized coordinate values obtained from surrogate models of static contact analyses (as in the standard FDK method [10]). With all kinematics defined, we use surrogate geometric models to compute the action of muscles on the multibody system [19,20]. With all loads and kinematics defined, we perform inverse dynamics and obtain generalized residual forces. We subtract any reserve actuator loads in design variable category (2) from the generalized residual forces to obtain the net residual loads. To satisfy the equations of motion, a set of nonlinear constraints is defined where the net residual loads must be approximately equal to zero.

2.3. Specific components

2.3.1. Modeling joint contact as static analyses

The most unique aspect of the proposed approach is the way joint contact is modeled. Existing approaches incorporate contact by adding to the original system components that directly apply loads to contacting bodies based on the spatial relationship between their articular surfaces (e.g., elastic contact models). With this approach, surrogate contact model inputs are poses, and the outputs are corresponding loads. In our approach, we model elastic contact as a set of neural networks that emulate the behavior of finite element contact simulations, where surrogate contact model inputs are combinations of poses and loads, and the outputs are the corresponding combinations of loads and poses. In this way, each surrogate contact model evaluation yields the solution to a static analysis, which is a unique way of performing surrogate contact modeling. For surrogate contact model inputs, the poses correspond to insensitive directions while the loads correspond to sensitive directions. We use our freely-available software Surrogate Contact Modeling Toolbox (SCMT) [21] to build our neural-network contact model approximations.

There are three advantages to treating contact interactions as static analyses. The first advantage is that this method allows us to more easily parametrize the optimization design space to avoid evaluating the surrogate model outside of its valid domain. When poses are used as surrogate contact model inputs for sensitive directions, it is easy for a simulation to choose pose values that correspond to unrealistic deeply penetrating or out of contact situations. The second advantage is that the optimization is not highly sensitive to any of the design variables, leading to a well-conditioned contact problem. In contrast, use of pose values in sensitive directions as design variables leads to poorly conditioned optimization problems. Finally, because we use surrogate models based on feedforward neural networks, we simultaneously speed up contact computation by several orders of magnitude and smooth out all discontinuities in the contact model.

We have created a simple analytical example problem to illustrate the approach and its advantages. This example can be found in the Supplementary Material.

2.4. Surrogate modeling of muscle-tendon lengths, velocities, and moment arms

To provide smooth and computationally fast evaluations of musculoskeletal geometry, we created a tool that constructs surrogate models of muscle-tendon lengths, velocities, and moment arms automatically. The tool adaptively samples muscle-tendon lengths and moment arms from an OpenSim [17] model and fits them using multivariable polynomial regression, producing surrogate musculoskeletal geometry models while keeping the root-mean-square of the relative muscle-tendon length and moment arm errors below a user-specified threshold.

In this approach, lengths are fit as either cubic, quartic, or quintic functions of the generalized coordinates on which the muscle path actuator depends. Each muscle-tendon length l is modeled as a function of the m generalized coordinates $q_1, q_2, \ldots, q_m$ that influence it:

$$l = f(q_1, q_2, \ldots, q_m) \tag{1}$$

Using the work-equivalence principle [22], we generate expressions for moment arms S_i for $i = 1, 2, \ldots, m$.

$$S_i = -\frac{\partial l}{\partial q_i} \tag{2}$$

Using the chain rule, we obtain expressions for the muscle-tendon velocity v.

$$v = \sum_{i=1}^{m} \left(\frac{\partial l}{\partial q_i} \frac{\partial q_i}{\partial t} \right) \tag{3}$$

$$v = -\sum_{i=1}^{m} S_i \dot{q}_i \tag{4}$$

We use a custom Hill-type muscle-tendon model with rigid tendon to compute the muscle forces F^M:

$$F^M = F_o^M \left[a f_l(\tilde{l}^M) f_v(\tilde{v}^M) + f_p(\tilde{l}^M) \right] \cos\alpha \tag{5}$$

Each muscle force is a function of its activation a, muscle-tendon length l, muscle-tendon velocity v, tendon slack length l_s^T, optimal fiber length l_o^M, maximum isometric muscle force F_o^M, and pennation angle α. The functions f_l, f_v, and f_p correspond to the normalized active force-length curve, active force–velocity curve, and passive force-length curve, which we modeled using smooth analytic functions [23], $\tilde{l}^M$ and $\tilde{v}^M$ correspond to the normalized muscle fiber length and normalized muscle fiber velocity. For more details about this process, see the Supplementary Material.

2.4.1. Computing parallelized skeletal inverse dynamics

At the core of our computational framework is a software interface that allows one to evaluate OpenSim multi-body dynamic models from within Matlab (The Mathworks, Natick, MA) with small overhead and with parallelization capability. This interface allows us to incorporate surrogate joint contact and musculoskeletal geometric models, incorporate custom muscle-tendon models, perform inverse dynamic and state derivative calculations, and obtain quantities of interest such as marker locations and reported forces. While OpenSim provides a Matlab SWIG interface, it was not appropriate for our purposes due to the relatively large overhead caused by its wrappers. Our solution was to use a Matlab mex file that places and retrieves data to and from shared memory (Windows mapped memory files), and a C++ program that runs in the background and scatters multiple queries over multiple processors using a Message Passing Interface (MPI) (Fig. 3). MPI not only has a small thread management overhead but is also the de-facto standard in high performance computing clusters and allowed us to perform queries of our force components and system dynamics using a heterogeneous network of computers.

3. Example applications: Muscle and knee contact force optimizations with surrogate contact models

We present two related example applications where we predict leg muscle forces, knee joint contact forces, and knee joint kinematics simultaneously during walking. The first example application is a static optimization (mod-FDK) and the second a dynamic optimization (direct collocation). Both applications use the same OpenSim musculoskeletal model of a pelvis and right leg implanted with a total knee replacement (TKR), the same tibiofemoral (TF) and patellofemoral (PF) surrogate contact models, and the same surrogate models of musculoskeletal geometry. For both example applications, secondary knee motions were solved in a quasi-static sense from surrogate contact models, meaning the generalized velocities and accelerations for these motions were set to zero during the solution process. Both applications also had the experimentally measured ground reactions applied directly to the foot segment of the model. The model and experimental gait data were taken from the first Grand Challenge Competition to Predict *In-Vivo* Knee Loads [24].

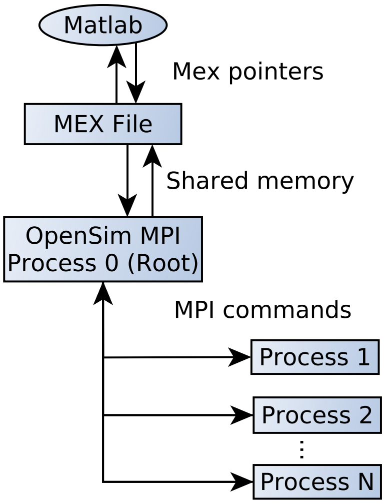

Fig. 3. Schematic of information flow for the custom Matlab-OpenSim interface. The scheme transfers data from the Matlab workspace to a separate program running MPI via a mex file and shared memory.

Table 1
Surrogate contact model inputs and outputs for a complete knee model where the contact problem is treated as a static analysis. For the tibiofemoral model, the inputs are the anterior-posterior translation, medial-lateral translation, internal-external rotation, and flexion angle, along with superior-inferior force and adduction moment. For the patellofemoral model, the inputs are the anterior-posterior and medial-lateral compressive forces, as well as superior-inferior translation and all rotations of the femoral component with respect to the patella. The outputs yield all corresponding loads and pose parameters which result from a static analysis.

	TF	PF
Inputs	t_x, ${}^{MB}F_y$, t_z, ${}^{MB}T_x$, r_y, r_z	${}^{MB}F_x$, t_y, ${}^{MB}F_z$, r_x, r_y, r_z
Outputs	$F_x, F_y^{med}, F_y^{lat}, F_z, T_x, T_y, T_z, r_x, t_y$	F_y, T_x, T_y, T_z, t_x, t_z

3.1. Surrogate contact models of static analyses

We built surrogate contact models from a data set of static contact analyses performed using the finite element software FEBio [25]. The coordinate system alignment and naming conventions for the inputs and outputs of the contact models are as follows: For the TF model, the x-axis points posteriorly, the y-axis superiorly, and the z-axis medially. For the PF model, the x-axis points posteriorly, the y-axis superiorly, and the z-axis medially. Forces and torques are represented as uppercase ${}^{body}F_{dir}^{side}$ and ${}^{body}T_{dir}^{side}$ while translations and rotations are represented with lowercase t_{dir} and r_{dir}. In this convention, *dir* refers to an associated spatial direction (x, y, or z), *side* can be medial (*med*), lateral (*lat*), or the net load (no superscript), and *body* can be the moving body (*MB*) or the fixed body (no superscript) on which the loads are applied.

The created surrogate contact models consisted of the following inputs and outputs (Table 1). The TF surrogate contact model was composed of nine neural networks that calculated the load

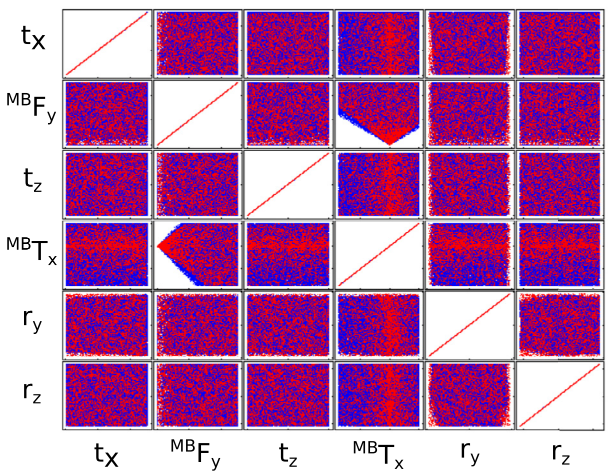

Fig. 4. Matrix scatterplot of the point cloud used to construct surrogate models of contact analysis for the TF joint. Blue dots represent the training data set and red dots represent generated scattered points throughout the re-parametrized space. Notice that the empty white space resulting from infeasible combinations of vertical force and adduction moment is excluded from the search space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

quantities $F_x, F_y^{med}, F_y^{lat}, F_z, T_x, T_y, T_z$ expressed in the fixed body (tibial component) as well as the sensitive poses t_y and r_x representing superior-inferior translation and adduction angle, respectively. The inputs to all neural networks in the TF contact model were t_x, ${}^{MB}F_y$, t_z, ${}^{MB}T_x$, r_y, r_z where ${}^{MB}F_y$ and ${}^{MB}T_x$ are the net superior-inferior force and adduction-abduction torque applied to the femoral component during the TF finite element static analyses used for generating surrogate model sample points. The PF surrogate model was composed of six neural networks which calculated the load quantities F_y, T_x, T_y, T_z expressed in the fixed body (patellar button) as well as the sensitive poses t_x, t_z. The inputs to all neural networks of the PF contact model were ${}^{MB}F_x$, t_y, ${}^{MB}F_z$, r_x, r_y, r_z where ${}^{MB}F_x$ and ${}^{MB}F_z$ are the compressive anterior-posterior force and medial-lateral force applied on the femoral component during the PF finite element static analyses used for generating surrogate model sample points.

To define feasible, physically realistic regions of inputs space, we analyzed the point clouds used to build the surrogate contact models using a scatter plot matrix of the input data, which is a set of projections of the input data onto its various dimensions. To bound the contact model inputs to the valid region of the input space, we fitted smooth functions to the boundaries of the point cloud projections. For the TF model, we bound the allowable vertical force range to a smooth linear-to-linear function of the adduction moment (Fig. 4). We followed a similar approach for the PF contact model. While this approach does not guarantee that the surrogate model will be valid, it does remove from the optimization design space large regions of invalid input space corresponding to non-physical configurations.

3.2. Surrogate musculoskeletal geometry

We used a custom program to automatically generate surrogate models of musculoskeletal geometry by sampling muscle-tendon lengths and moment arms from our subject-specific OpenSim model in a wide range of poses that spanned beyond those of the walking motion to be simulated. We specified a maximum RMS error of 10% in the moment arms relative to the muscle's moment arm range, which was achieved by sampling points and changing polynomial order adaptively.

3.3. Ligament models

We used a point-to-point spring to model the patellar tendon and coordinate springs to model stiffness in the tibiofemoral joint. The coordinate springs were not meant to be accurate representations of ligament actions but only to limit the motion of the contacting bodies. Neither the resting length nor the stiffness of the coordinate springs was calibrated for either approach. We used low stiffness springs for the mod-FDK approach to minimize simulated ligament action and higher stiffness springs for the direct collocation approach since the optimization would not converge to a solution with low stiffness springs.

3.4. Example 1: Modified force-dependent kinematics (mod-FDK)

The mod-FDK approach is a method for performing static optimization with deformable joint contact. Just as with standard FDK [10], this approach performs one optimization per time frame, and the secondary joint kinematics and contact loads are solved together with the muscle activations. The cost function minimized a weighted sum of squares of muscle activations, reserve actuator loads, and contact forces:

$$\begin{aligned} \underset{\boldsymbol{e},\boldsymbol{r},\boldsymbol{c}}{\text{minimize}} \quad & \sum_{i=1}^{n_e} e_i^2 + \sum_{i=1}^{n_r} w_i r_i^2 + w_c\left[(F_y^{med})^2 + (F_y^{lat})^2\right] \\ \text{subject to} \quad & \left[F^{res}_{(\boldsymbol{e},\boldsymbol{r},\boldsymbol{c})}\right]^2 - tol^2 < 0 \\ & {}^{TF}r_z - {}^{PF}r_z - \text{offset} < 0 \end{aligned} \tag{6}$$

In the above problem formulation, the design variables were muscle activations e, reserve actuator activations r, and surrogate contact model inputs c (review Table 1 and Fig. 5). The cost due to reserve actuation and contact forces is weighed with values chosen to minimize reserve actuator action and to scale the cost related to contact forces. F^{res} stands for the vector of residual loads, which contains the unbalanced generalized force for each degree of freedom. We imposed near zero residual load tolerance for all degrees of freedom except hip internal rotations, which our model was unable to balance well. ${}^{TF}r_z$ represents the tibiofemoral flexion coordinate and ${}^{PF}r_z$ the patellofemoral flexion coordinate. The inequality constraint enforcing an offset between tibiofemoral and patellofemoral flexion angles helped the optimizer avoid local minima in non-physiological configurations. We used an offset value of 20 degrees. We solved this problem using Matlab's fmincon SQP optimizer.

3.5. Example 2: Dynamic optimization using direct collocation

As opposed to the mod-FDK approach in the previous section, this approach performs a single optimization taking all time frames into account simultaneously. The direct collocation problem was set up to use implicit inverse dynamics as opposed to explicit forward dynamics while using jerk as an additional control:

$$\begin{aligned} \underset{\boldsymbol{e},\boldsymbol{r},\boldsymbol{c}}{\text{minimize}} \quad & \int_{t_0}^{t_f}\left[\frac{1}{n_e}\sum_{i=1}^{n_e} e_{i(t)}^2 + \frac{1}{n_r}\sum_{i=1}^{n_r} r_{i(t)}^2 + \frac{1}{n_j}\sum_{i=1}^{n_j} j_{i(t)}\right]dt \\ \text{subject to} \quad & \begin{bmatrix}\dot{q}\\ \dot{v}\\ \dot{a}\end{bmatrix} = \begin{bmatrix} v\\ a\\ j\end{bmatrix} \end{aligned} \tag{7}$$

$$F^{res}_{(\boldsymbol{e},\boldsymbol{r},\boldsymbol{c})} = 0$$

$${}^{TF}r_z - {}^{PF}r_z - \text{offset} < 0$$

$$\parallel (\mathbf{p}^{\text{exp}})_i - \left(\mathbf{p}^{\text{model}}\right)_i \parallel - tol_i = 0,\ i = 1, 2, \ldots, n_m.$$

The design variables were identical to those in the Mod-FDK approach except that they varied across time. The cost functional

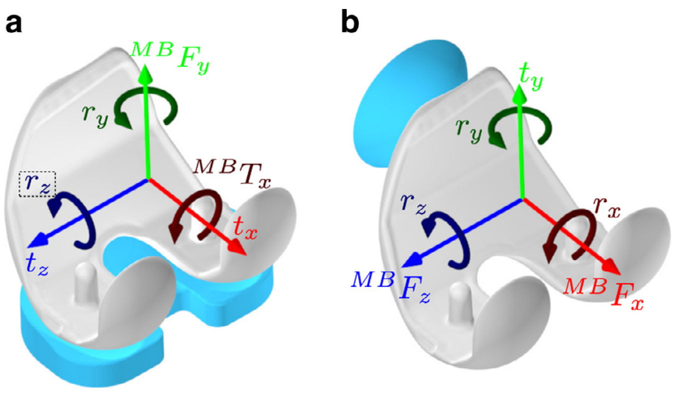

Fig. 5. Illustration of the design variables corresponding to the surrogate contact model inputs where t is translation, r is rotation, F is force, and T is torque. (a) Design variables corresponding to the tibiofemoral contact model inputs. Notice the r_z coordinate corresponding to knee flexion is only a design variable in the direct collocation approach while it is prescribed in the mod-FDK approach. (b) The design variables corresponding to patellofemoral contact model inputs.

minimized a combination of muscle activations $e_{i(t)}$, reserve actuator activations $r_{i(t)}$, and generalized jerk $j_{i(t)}$. Variables $q_{i(t)}$ represent only the primary generalized coordinates, $v_{i(t)}$ their first time derivatives, and $a_{i(t)}$ their second time derivatives. The first constraint enforces kinematic consistency, i.e., one curve is the derivative of another, thereby satisfying the dynamics of the system. The second and third constraints are similar to those in the Mod-FDK formulation, while the last constraint imposes bounds on the marker errors. In the last constraint, $(\mathbf{p}^{\text{exp}})_i$ stands for the position of the ith experimental marker with respect to the origin of the lab coordinate system, and $(\mathbf{p}^{\text{model}})_i$ stands for the corresponding position of the ith model marker.

We solved a discretized approximation of the problem using the general purpose optimal control solver GPOPS-II [26] with sparse linear solver IPOPT [27] in first derivative mode. Marker errors were limited to 5 mm for shank and foot markers and 10 mm for pelvis markers.

3.6. Results

The mod-FDK approach converged to a solution in 1 to 10 s of CPU time for each time frame while the direct collocation method took 170 s to converge using 60 collocation points when starting from a good previous solution as an initial guess. Both problems were solved on a PC workstation with 16 processors clocking 3.4 GHz each.

Both approaches succesfully calculated tibiofemoral and patellofemoral contact forces and secondary knee kinematics. The predicted knee kinematics were realistic (Fig. 6) and the contact forces for both approaches compared reasonably well to the experimental contact forces (Fig. 7). Both methods overestimated the second peak in medial contact force but the direct collocation method severely underestimated the peak lateral contact force. Interestingly, both methods yielded similar net patellar contact forces, which reached approximately 400 N and 200 N for the first and second peak, respectively.

The computed secondary knee kinematics were influenced by the different coordinate spring stiffness values used for the two approaches. Much larger anterior-posterior translation was predicted by the mod-FDK approach, which used a set of low stiffness springs (Fig. 8). The superior-inferior translation for both methods approximated the expected trend from fluoroscopy data. The flexion angle was similar for both methods but different from the experimental flexion angle obtained from fluoroscopy data corresponding to a different gait cycle.

To quantify the benefits of using parallel processing and surrogate musculoskeletal geometry, we performed a benchmark test. We used our pelvis-leg OpenSim model to evaluate the CPU time required to perform inverse dynamics for 100 collocation points. We evaluated our custom C++ interface having OpenSim's Matlab SWIG interface as baseline. Our C++ interface produced a speedup of 19x. Enabling paralellization increased the speedup to 115x. Enabling parallelization and incorporating surrogate musculoskeletal geometry together produced a speedup of 408x (Table 2). These benchmark tests were performed on a workstation with two Intel Xeon processors, using 13 out of 16 available physical cores.

We tested our surrogate models of static contact analysis for both accuracy and computational speed. The models were computationally inexpensive with the TF model and PF model requiring an average of 147 and 83 µs per evaluation, respectively. The

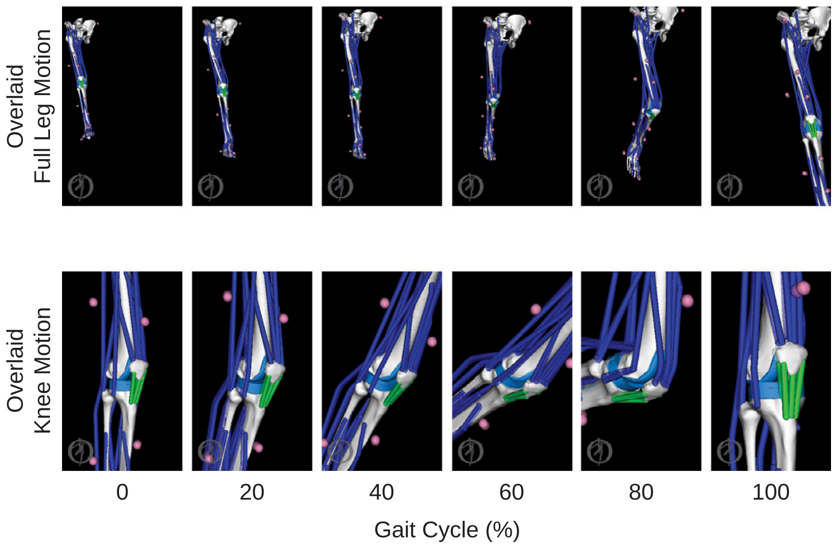

Fig. 6. Animation strips of model and knee kinematics resulting from mod-FDK and direct collocation approaches overlaid. The gross motion of the models are very similar with few notable differences in knee flexion and patella rotation. Plots comparing the most relevant kinematics can be found in Fig. 8.

Table 2
CPU times and speedups computed for 100 collocation points using parallelization and surrogate muscle geometry.

Interface	Parallelized	Time (ms)	Speedup
OpenSim's Matlab interface	No	1083.5	1
C++ without surrogate muscle geometry	No	57.0	19
	Yes	9.4	115
C++ with surrogate muscle geometry	No	19.1	57
	Yes	2.7	408

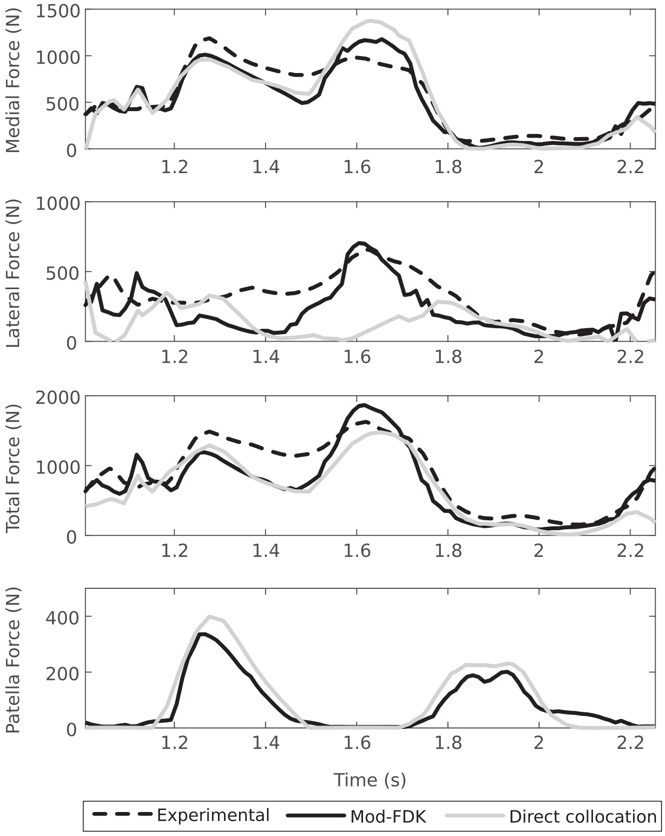

Fig. 7. Knee contact forces calculated using mod-FDK (static optimization) and direct collocation (dynamic optimization) compared to experimental tibiofemoral forces measured *in-vivo* [24].

accuracy results showed RMS translation errors below 0.2 mm and orientation errors below 0.1 degrees. Force errors were below 15 N, and the largest torque error was below 0.7 N mm (see Supplementary Material for details).

4. Discussion

We have presented a novel computational approach for estimating muscle forces, joint contact forces, and joint kinematics with low computational cost, improved gradients, and all sources of non-smoothness removed. We moved most of the computational expense into a pre-processing stage where surrogate models of contact forces and musculoskeletal geometry were trained and saved for later use. We also developed an efficient interface between the optimizer and the OpenSim multibody dynamics engine that lets us use parallelization with little overhead. We have shown that joint models with elastic contact can be incorporated into static and dynamic optimizations of human movement, and that problems formulated within this framework can converge in times ranging from seconds for single time frames to minutes for an entire motion cycle.

The most novel feature of this framework is the introduction of neural network-based surrogate models that efficiently emulate the behavior of static contact analyses within the optimization's nonlinear constraints. This aspect renders the nonlinear constraint function insensitive to measures such as contact penetration. It also allows us to couple complex finite element models of elastic or hyperelastic contact with body-scale or limb-scale biomechanical optimizations, addressing a challenging multi-scale modeling problem.

The results obtained in our example applications not only demonstrate the feasibility of the presented framework but also make the case for the addition of carefully calibrated ligament models, as calculated joint kinematics were greatly affected by ligament stiffness. Moreover, because of its computational speed, our approach could potentially be used to tune ligament model parameter values in an outer optimization such that the predicted joint kinematics matched fluoroscopic measurements. However, ligament model creation and calibration is beyond the scope of the present study.

The example applications yielded reasonable contact force predictions. To obtain these results, we had to adjust optimal fiber lengths such that muscles exerted little passive force during the gait cycle. We also made sure that the model was able to match experimentally obtained medial and lateral knee contact forces before performing the predictive optimizations.

The mod-FDK approach has the advantage of being more robust to problem formulation than is the optimal control approach. In our experience, the direct collocation approach is more challenging as many problem formulations fail to converge, and it is difficult to track down where the problem lies. For example, when we added a contact force minimization term to the direct collocation approach, the optimization would not converge. While performing an optimization based on direct collocation to predict motion without constraints on markers is theoretically possible, we suspect much “tweaking” would be required to achieve convergence.

Despite obtaining improved optimization performance, the mod-FDK approach suffered from multiple local minima, which forced us to perform multiple optimizations from different initial guesses. We also found that the passive forces generated by Hill-type muscle models were insufficient to cause a 6-DOF patella to remain within the trochear groove of the femoral component in the absence of additional soft tissue forces. For this reason, we modeled the patellar tendon as a point-to-point spring, which can be loaded in both tension and compression. In both examples, our model was unable to balance the hip internal rotation moment, adduction moment, and flexion moment simultaneously, which is why we removed the internal rotation moment as an inverse dynamic constraint. We believe this issue was caused by modeling errors in the attachment points of hip muscles in our OpenSim musculoskeletal model.

Comparing our framework to other approaches with slow and discontinuous functions is not realistic for two reasons. First, performing gradient-based optimizations on non-smooth functions is

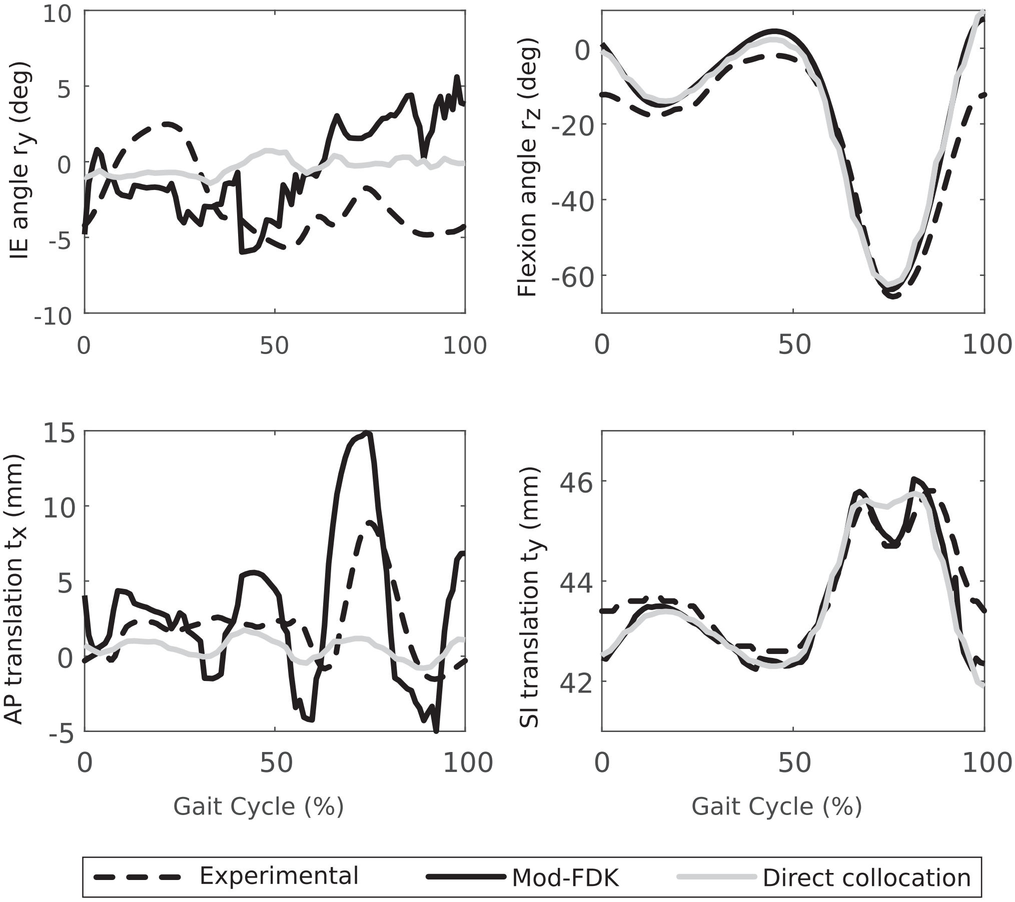

Fig. 8. Knee kinematics computed via mod-FDK and Direct collocation compared to kinematics measured using fluoroscopy during a different gait cycle. The anterior-posterior translation and internal-external rotation have large errors due to the use of coordinate springs instead of an anatomic ligament model.

fundamentally incorrect and leads to unpredictable results. Second, we could not perform the same optimizations with related models possessing slow functions since in our formulations, computational smoothness and computational speed are tied together through our surrogate modeling approaches.

Our framework possesses several important limitations. The first limitation is in the method we used to define bounds on the contact design variables. While our approach is relatively simple, relying on the projections of the point cloud to classify valid regions of the design space is crude and not guaranteed to yield acceptable accuracy. Our approach could be improved by implementing a model that assesses design space validity automatically. A second limitation is the use of synchronous parallelization in our Matlab-OpenSim interface, which works well for direct collocation when multiple queries are made at once but could not be used to perform multiple static optimizations in parallel. Implementing a dual synchronous/asynchronous parallelization approach would make the interface suitable for both static and dynamic optimizations. The third limitation is that the velocities and accelerations of the secondary kinematics are assumed to be zero in our modified FDK and direct collocation approaches, as in the FDK approach. However, we have found this approximation to be reasonable since in our experience adding secondary kinematics to the state vector in dynamic optimizations only yields vibration artifacts resulting from the highly stiff ligament and contact models. A fourth limitation involves using multivariable polynomials to model muscle-tendon lengths, velocities, and moment arms. While this approach is relatively simple as surrogate model training can be done via regression, it has two problems. The first problem is limited accuracy, which we addressed by using higher order polynomials and adaptive building, though our custom tool currently limits muscle-tendon length polynomials to at most fifth degree. This limitation could potentially be overcome by using a different surrogate modeling approach such as B-splines [28] or neural networks. The second problem is that some muscles may depend on many DOFs. Since the number of sample points required to maintain surrogate model accuracy scales exponentially with the number of input dimensions, when a muscle's geometric path is dependent on many DOFs, the limited training data set becomes sparse and the surrogate model of musculoskeletal geometry loses accuracy. A final limitation is that the modeled joints are assumed to remain in contact always. Allowing liftoff would make the elastic contact model non-invertible, which the presented approach cannot handle. However, we believe surrogate models of static contact analyses could be coupled with complementarity constraints to allow for contact liftoff in optimizations without introducing discontinuities.

The computational framework presented in this study allows for joint contact models to be added to both static and dynamic gradient-based muscle force optimizations with efficiency and computational speed, allowing muscle forces, elastic contact forces, secondary knee kinematics, and even primary joint kinematics to be calculated simultaneously using optimization methods. Our approach allows for the explicit mechanical coupling between the joint of interest and the rest of the body, which opens up new opportunities in biomechanical simulation.

Competing interests

The authors have no competing interests.

Acknowledgments

Funding for this study was provided by NIH grant R01EB009351, NSF grant CBET 1404767, and the University of Florida.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.medengphy.2018.02.002.

References

[1] Moissenet F, Chèze L, Dumas R. Anatomical kinematic constraints: consequences on musculo-tendon forces and joint reactions. Multibody Syst Dyn 2012;28(1–2):125–41. doi:10.1007/s11044-011-9286-3.

[2] Taylor WR, Heller MO, Bergmann G, Duda GN. Tibio-femoral loading during human gait and stair climbing. J Orthop Res 2004;22(3):625–32. doi:10.1016/j.orthres.2003.09.003.

[3] Shelburne KB, Torry MR, Pandy MG. Contributions of muscles, ligaments, and the ground-reaction force to tibiofemoral joint loading during normal gait. J Orthop Res 2006;24(10):1983–90. doi:10.1002/jor.20255.

[4] Kim HJ, Fernandez JW, Akbarshahi M, Walter JP, Fregly BJ, Pandy MG. Evaluation of predicted knee-joint muscle forces during gait using an instrumented knee implant. J Orthop Res 2009;27(10):1326–31. doi:10.1002/jor.20876.

[5] Walter JP, Korkmaz N, Fregly BJ, Pandy MG. Contribution of tibiofemoral joint contact to net loads at the knee in gait. J Orthop Res 2015:n/a. doi:10.1002/jor.22845. http://www.ncbi.nlm.nih.gov/pubmed/25676012.

[6] Lin Y-C, Walter JP, Banks Sa, Pandy MG, Fregly BJ. Simultaneous prediction of muscle and contact forces in the knee during gait. J Biomech 2010;43(5):945–52. doi:10.1016/j.jbiomech.2009.10.048. http://www.ncbi.nlm.nih.gov/pubmed/19962703.

[7] Thelen DG, Choi KW, Schmitz AM. Co-simulation of neuromuscular dynamics and knee mechanics during human walking. J Biomech Eng 2014;136(2):0210331. doi:10.1115/1.4026358. http://biomechanical.asmedigitalcollection.asme.org/article.aspx?articleid=1812595 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4023657&tool=pmcentrez&rendertype=abstract.

[8] Smith CR, Vignos MF, Lenhart RL, Kaiser J, Thelen DG. The influence of component alignment and ligament properties on tibiofemoral contact forces in total knee replacement. J Biomech Eng 2016;138(2):021017. doi:10.1115/1.4032464. http://www.ncbi.nlm.nih.gov/pubmed/26769446 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4844247.

[9] Marra MA, Vanheule V, Fluit R, Koopman BHFJM, Rasmussen J, Verdonschot N, et al. A subject-specific musculoskeletal modeling framework to predict *in vivo* mechanics of total knee arthroplasty. J Biomech Eng 2015;137(2):020904. doi:10.1115/1.4029258. http://www.ncbi.nlm.nih.gov/pubmed/25429519 http://biomechanical.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4029258.

[10] Andersen MS, Damsgaard M, Rasmussen J. Force-dependent kinematics : a new analysis method for non-conforming joints. In: Proceedings of XIII international symposium on computer simulation in biomechanics. Leuven, Belgium; 2011.

[11] Skipper Andersen M, de Zee M, Damsgaard M, Nolte D, Rasmussen J, Andersen MS, et al. Introduction to force-dependent kinematics : theory and application to mandible modeling. J Biomech Eng 2017;139(c):091001. doi:10.1115/1.4037100. http://biomechanical.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4037100.

[12] Marra MA, Andersen MS, Damsgaard M, Koopman BFJM, Janssen D, Verdonschot N. Evaluation of a surrogate contact model in force-dependent kinematic simulations of total knee replacement. J Biomech Eng 2017;139(August):081001. doi:10.1115/1.4036605. http://www.ncbi.nlm.nih.gov/pubmed/28462424.

[13] Guess TM, Stylianou AP, Kia M. Concurrent prediction of muscle and tibiofemoral contact forces during treadmill gait. J Biomech Eng 2014;136(2):021032. doi:10.1115/1.4026359. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4023658&tool=pmcentrez&rendertype=abstract.

[14] Moissenet F, Chèze L, Dumas R. A 3D lower limb musculoskeletal model for simultaneous estimation of musculo-tendon, joint contact, ligament and bone forces during gait. J Biomech 2014;47((1):50–8). doi:10.1016/j.jbiomech.2013.10.015. http://www.ncbi.nlm.nih.gov/pubmed/24210475 http://www.sciencedirect.com/science/article/pii/S0021929013004697.

[15] Fregly BJ, Banks SA, D'Lima DD, Colwell CW. Sensitivity of knee replacement contact calculations to kinematic measurement errors. J Orthop Res 2008;26(9):1173–9. doi:10.1002/jor.20548.

[16] Delp SL, Loan J. A graphics-based software system to develop and analyze models of musculoskeletal structures. Comput Biol Med 1995;25(1):21–34. doi:10.1016/0010-4825(95)98882-E. http://linkinghub.elsevier.com/retrieve/pii/001048259598882E.

[17] Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, et al. OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng 2007;54(11):1940–50. doi:10.1109/TBME.2007.901024. http://ieeexplore.ieee.org/document/4352056/.

[18] Damsgaard M, Rasmussen J, Christensen ST, Surma E, de Zee M. Analysis of musculoskeletal systems in the anybody modeling system. Simul Model Pract Theory 2006;14(8):1100–11. doi:10.1016/j.simpat.2006.09.001. http://linkinghub.elsevier.com/retrieve/pii/S1569190X06000554.

[19] Menegaldo LL, de Toledo Fleury A, Weber HI. Moment arms and musculotendon lengths estimation for a three-dimensional lower-limb model. J Biomech 2004;37(9):1447–53. doi:10.1016/j.jbiomech.2003.12.017. http://www.ncbi.nlm.nih.gov/pubmed/15275854 http://linkinghub.elsevier.com/retrieve/pii/S0021929003004809.

[20] Van Den Bogert AJ, Blana D, Heinrich D. Implicit methods for efficient musculoskeletal simulation and optimal control. Proc IUTAM 2011;2:297–316. NIHMS150003. doi:10.1016/j.piutam.2011.04.027.

[21] Eskinazi I, Fregly BJB. Surrogate modeling of deformable joint contact using artificial neural networks. Med Eng Phys 2015;37(9):885–91. doi:10.1016/j.medengphy.2015.06.006. http://www.sciencedirect.com/science/article/pii/S135045331500154X.

[22] Sherman MA, Seth A, Delp SL. What is a moment arm? Calculating muscle effectiveness in biomechanical models using generalized coordinates. In: Proceedings of the ASME design engineering technical conferences, Vol. 2013. NIH Public Access; 2013. https://doi.org/10.1115/DETC2013-13633. http://www.ncbi.nlm.nih.gov/pubmed/25905111 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4404026.

[23] De Groote F, Kinney AL, Rao AV, Fregly BJ. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. Ann Biomed Eng 2016;44(10):2922–36. doi:10.1007/s10439-016-1591-9. http://link.springer.com/10.1007/s10439-016-1591-9.

[24] Fregly BJ, Besier TF, Lloyd DG, Delp SL, Banks SA, Pandy MG, et al. Grand challenge competition to predict *in vivo* knee loads. J Orthop Res 2012;30(4):503–13. doi:10.1002/jor.22023. http://www.ncbi.nlm.nih.gov/pubmed/22161745 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4067494.

[25] Maas SA, Ellis BJ, Ateshian GA, Weiss JA. FEBio: finite elements for biomechanics. J Biomech Eng 2012;134(1):011005. doi:10.1115/1.4005694. https://pdfs.semanticscholar.org/2836/af6e2b8bfb0dd6e271ac9b24e9b3715aa56d.pdf http://biomechanical.asmedigitalcollection.asme.org/article.aspx?articleid=1431396.

[26] Patterson MA, Rao AV. GPOPS II : a MATLAB software for solving multiple-phase optimal control problems using HP adaptive gaussian quadrature collocation methods and sparse nonlinear programming. ACM Trans Math Softw 2013;39(3):1–38. doi:10.1145/0000000.0000000. http://dl.acm.org/citation.cfm?doid=2684421.2558904.

[27] Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 2006;106(1):25–57. doi:10.1007/s10107-004-0559-y.

[28] Sartori M, Reggiani M, van den Bogert AJ, Lloyd DG. Estimation of musculotendon kinematics in large musculoskeletal models using multidimensional B-splines. J Biomech 2012;45(3):595–601. doi:10.1016/j.jbiomech.2011.10.040.