Abstract

We present a biped locomotion controller for humanoid models actuated by more than a hundred Hill-type muscles. The key component of the controller is our novel algorithm that can cope with step-based biped locomotion balancing and the coordination of many nonlinear Hill-type muscles simultaneously. Minimum effort muscle activations are calculated based on muscle contraction dynamics and online quadratic programming. Our controller can faithfully reproduce a variety of realistic biped gaits (e.g., normal walk, quick steps, and fast run) and adapt the gaits to varying conditions (e.g., muscle weakness, tightness, joint dislocation, and external pushes) and goals (e.g., pain reduction and efficiency maximization). We demonstrate the robustness and versatility of our controller with examples that can only be achieved using highly-detailed musculoskeletal models with many muscles.

1 Introduction

Reproducing realistic human locomotion in physically based simulation has been a long-standing goal of computer graphics research. Many biped controllers in computer graphics assumed a linked structure of rigid bodies connected by idealized joints that can generate arbitrarily large torques along any directions immediately whenever needed. Such simplified body and actuation models made balance control and trajectory tracking plausible for full-body locomotion and even for acrobatic full-body actions. Recently, there have been continuous efforts to simulate and control musculoskeletal models with many muscles. The human body has over 700 skeletal muscles and 200 of them are especially important for locomotion because they move large bones. Harmonious coordination of many muscles results in complex human movements. Designing a control law for a many-muscle actuated humanoid poses several challenges: underdetermined control systems, the complexity of muscle contraction dynamics, and integrated controller design. The coordination of many muscles is inherently underdetermined since there are more muscles than the number of body degrees of freedom to actuate. Multiple sets of muscle actuations can lead to the same set of joint torques, and thus the same resulting motion. Moreover, it is often unclear which motion is best suited to given tasks under various intentions or conditions. The best one should be determined in a given situation. Secondly, the contraction dynamics of each individual muscle is a highly nonlinear process, which is often modeled using a three-element Hill-type model. The torque generated by an individual muscle depends on many factors such as the geometry of joints and bones, the level of muscle activation, muscle length, and the velocity of muscle contraction. It is often impossible to generate an exact torque immediately, which can only be achieved through a dynamic process. Thirdly, due to the complexity of muscle dynamics, it is difficult to simply replace idealized joints with muscle-based actuators in existing biped controllers. Instead, the controllers should be redesigned to integrate fully functioning muscle dynamics.

In this paper, we present a new muscle-actuated biped controller that scales well to cope with highly detailed musculoskeletal models having more than a hundred musculotendon actuators. In biomechanics, such highly detailed musculoskeletal models have often been used for static analysis of human movements, but have not been employed to design locomotion controllers. Our controller is equipped with a step-based balance mechanism to reproduce a variety of human gait patterns, ranging from low-energy normal walk to highly energetic quick steps, while being resilient to external perturbations.

Our controller has two major technical components, muscle optimization and trajectory optimization, which integrate muscle actuators seamlessly into an existing step-based feedback controller. Given a reference motion of arbitrary gait patterns, muscle optimization calculates the optimal coordination of muscle activations to track the reference motion while resolving actuation redundancy by minimizing efforts. Since muscle actuations are computed on a per-frame basis, the solution is optimal only instantaneously. Our trajectory optimization actively modulates the full cycle (half cycle if the gait is symmetric) of the reference motion so that muscle dynamics can be accounted for in a longer horizon. The reference motion is optimized at the preprocessing phase and the user can choose to reproduce the reference motion as closely as possible or adapt the motion to meet new conditions and intentions at runtime simulation.
We demonstrate the robustness and applicability of our controller using three musculoskeletal models that walk and run in many different gaits. Each individual gait can be adapted to new body conditions (e.g., muscle weakness, tightness, dislocated joints, and external pushes) or new objectives (e.g., pain reduction and efficiency maximization). Our many-muscle controller can reproduce the subtle nuances of pathologic gait conditions that match real patient data.

## 2 Related Work

Controlling biped locomotion in physically based simulation has been studied in computer graphics and robotics for several decades. Many controllers are equipped with finite state machines and feedback rules for trajectory shaping and balance control [Hodgins et al. 1995; Yin et al. 2007]. Some others employed simplified dynamic models abstracting the human body to mitigate the complexity of the full-body model [Kwon and Hodgins 2010; Mordatch et al. 2010; Ye and Liu 2010; Coros et al. 2010; Tsai et al. 2010]. Motion capture data have also been employed to improve the naturalness of simulated gaits [Sok et al. 2007; da Silva et al. 2008; Muco et al. 2009; Lee et al. 2010; Kwon and Hodgins 2010].

Optimization techniques have been used to facilitate robust controller design in several different ways. Given a baseline controller, its parameters can be optimized to adapt to unexpected disturbances or changes of terrains [Wang et al. 2010; Wu and Popović 2010]. Liu et al. [2012] optimized an affine feedback policy to perform parkour-style terrain runs. Per-frame optimization of instantaneous control signals is usually formulated as a quadratic programming problem, which can be solved efficiently in an online manner [Abe et al. 2007; da Silva et al. 2008; Macchietto et al. 2009; de lasa et al. 2010; Wu and Popović 2010; Brown et al. 2013]. Optimizing reference trajectories for physically simulated characters has been applied to various human and animal locomotion control or simulation [Sok et al. 2007; Tan et al. 2011; Al Borno et al. 2013].

Many researchers in computer graphics have designed musculoskeletal models and simulate them in physically based simulation. There have been muscle models designed for specific body parts, for example, the torso [DiLorenzo et al. 2008], neck-head-face [Lee and Terzopoulos 2006], hand [Sueda et al. 2008], face [Sifakis et al. 2005] and upperbody [Teran et al. 2005; Lee et al. 2009]. These models are actuated by muscle forces to generate anatomically and physically realistic simulated motions. A simplified form of Hill-type model has been used to determine actuation forces based only on the force-length relations of muscle fibers ignoring the time-derivatives of muscle lengths [Sifakis et al. 2005; Sueda et al. 2008]. Lee et al. used another form of simplified Hill-type model using a linearized force-length-velocity curve [Lee and Terzopoulos 2006; Lee et al. 2009]. Their coactivation control method exploits the linearity to resolve the redundancy of muscle activations approximately. A more realistic nonlinear force-length-velocity curve is used in torso simulation [DiLorenzo et al. 2008]. de Sapio et al. [2005] discussed operational-space feedback control for musculoskeletal simulation using a nonlinear force-length-velocity curve. These studies assumed musculotendon actuators with constant-length tendons or no tendon at all to reduce the complexity of the muscle dynamics model. Several studies used volumetric representations of human muscles to improve appearance or modeling accuracy [Sifakis et al. 2005; Teran et al. 2005; Lee et al. 2009].

Biomechanics researchers have studied the simulation of various human activities, such as vertical jumps, walks, and pedaling. Anderson and Pandy [1999; 2001] computed excitation trajectories for jumping and walking by formulating the problem as an optimal control problem, often called dynamic optimization in the biomechanics community. The excitations for a half cycle of locomotion were simultaneously optimized to minimize the metabolic energy consumption per unit moving distance. Thelen and Anderson [2006] computed excitation trajectories to simulate walking by performing optimizations only at the current time instance. In biomechanics, optimization approaches are often classified into two categories: dynamic optimization and static optimization, which roughly corresponds to space-time optimization and online optimization commonly used in computer graphics. Our muscle optimization is similar to static optimization in the sense that control signals are optimized at every time steps. A fundamental difference is that these biomechanics approaches aim at finding a single optimal trajectory performing a specific task, and lacks the ability to respond to unexpected disturbances. Our goal is to find an optimal strategy having the ability to interactively adapt.

Recently, graphics researchers have explored the simulation of muscle-actuated bipedal locomotion with Hill-type muscle models. Wang et al. [2012] presented a locomotion controller for a simplified musculoskeletal model, which has eight muscles on each leg. The muscle forces were applied only at the degrees of freedom on the sagittal plane. Geijtenbeek et al. [2013] presented another muscle-based controller for various bipedal creatures, in which both muscle routing and control parameters are optimized simultaneously. Mordatch et al. [2013] proposed a trajectory optimization method to generate walking, running and kicking motions actuated by muscles. These studies commonly used simplified musculoskeletal models and metabolic energy expenditure to measure efforts.

Although we use several technical components that are already known (e.g., the online quadratic programming, Covariance Matrix Adaptation, and step-based balancing mechanism), our contribution is presenting a particular combination of those components that works well with a large-scale problem of many-muscle locomotion control. Our controller can control highly detailed musculoskeletal models, reproduce a number of gait patterns, adapt to a wide range of body conditions and optimization objectives. This allows the controller to capture, generate, and adapt detailed nuances of biped gait patterns so that it can predictively simulate a variety of pathologic gaits under intentional changes of body conditions.

## 3 Humanoid Models

Our humanoid models are based on public-domain musculoskeletal models in the OpenSim file format [Delp et al. 2007]. We use three humanoid models in our experiments having 25 to 39 degrees of freedom (DOFs) and 62 to 120 muscles. Two of them are without arms as shown in Figure 1 (see Appendix A for details). The height and mass of all models are about 180 cm and 75 kg. All DOFs of the models are actuated by Hill-type musculotendon actuators [Zajac 1989], which are commonly used in biomechanics research (Figure 2).

A musculotendon actuator generates force depending on its activation level and other internal states. The generated force is transmitted to bones through the attachment points of the musculotendon actuator. In the remainder of the paper, we use the term “muscle” to indicate “musculotendon” or “musculotendon actuator” for convenience, except for being used in “muscle fiber”.

### 3.1 Muscle Force Generation

The contraction dynamics equation of the Hill-type muscle model describes the behavior of the three components of a musculotendon actuator: a serial element, a contractile element, and a parallel...
element (Figure 2). The contractile element generates active force parallel to the muscle fiber, respectively.

Figure 1: Three musculoskeletal models. The gait2562 model on the left has 25 DOFs and 62 muscles, the gait2592 model at the middle has 25 DOFs and 92 muscles, and the fullbody model on the right has 39 DOFs and 120 muscles.

Figure 2: The Hill-type muscle model is composed of a serial element (SE), a contractile element (CE), and a parallel element (PE), which represents the tendon, the muscle fiber, and the elastic material parallel to the muscle fiber, respectively.

The relationship between these forces is described as:

\[ f_{at} = f_t = f_m \cdot \cos(\alpha) = (f_{ce} + f_{pe}) \cos(\alpha), \]

where \( f_{at} \) is the musculotendon actuator force, \( f_t \) is the force acting on the muscle fiber, and \( \alpha \) is the pennation angle which is the angle between the tendon and the muscle fiber.

We can compute the forces of each component as follows:

\[ f_t = g_v(l_t), \]

\[ f_{ce} = a \cdot g_{al}(l_m) \cdot g_v(\dot{l}_m), \]

\[ f_{pe} = g_{po}(l_m), \]

where \( g_v, g_{al}, g_{po}, \) and \( g_v \) are the force-length relationship of the tendon, the active and passive force-length relationship and the velocity-force relationship of the muscle fiber, respectively. Here \( l_t, l_m, \) and \( a \) are the length of the tendon, the length and velocity of the muscle fiber, and the activation level. Note that \( l_t = l_{at} - l_m \cos(\alpha) \), where \( l_{at} \) is the length of the musculotendon actuator which is determined based on the current pose of the humanoid. We use a modified version of the contraction dynamics equation proposed by Thelen [2003]. (see Appendix B for more details about how to calculate \( g_v, g_{al}, g_{po}, g_v, \) and \( g_v^{-1} \).)

By rearranging Equation (1) (2), (3) and (4) and employing a passive damping term, muscle force \( f_m \) and the time-derivative of the length of muscle fiber \( \dot{l}_m \) can be obtained:

\[ f_m = a \cdot g_{al}(l_m) \cdot g_v(\dot{l}_m) + g_{po}(l_m) + b \cdot \dot{l}_m, \]

\[ \dot{l}_m = g_v^{-1}(a, l_m, \dot{l}_m), \]

where \( b \) is a damping coefficient that is set to 0.05 for all muscles. This nonzero damping term is practically useful to make the simulation stable because without it Equation (6) has a singularity issue when \( a \) is close to zero [Schutte 1993]. Our system updates the muscle fiber length \( l_m \) by integrating \( \dot{l}_m \) numerically. The initial values of \( a \) and \( l_m \) are provided as described in Appendix C.

3.2 Muscle Force Transfer

In the human body, each muscle has two ends attached to bones at the origin and the insertion, which form points, lines or areas. Our humanoid model simplifies origins and insertions as attachment points assuming all muscles are thin. A muscle transfers its force to the bones through these attachment points or contact points with other bones or muscles. Contact points are implemented using conditional contact points that simulate the generation and elimination of contacts depending on the joint configurations [Delp et al. 2007] (Figure 3).

Let \( b^i \) denote the \( i^{th} \) bone of the model, \( m^j \) be the \( j^{th} \) muscle, \( P^j \) be the position of the \( k^{th} \) attachment or contact point of \( m^j \) in the global coordinate, \( b^j_k \) be the bone on which \( P^j \) is attached or contacted. Here \( P^j \) is indexed such that \( P^j_{k+1} \) is an adjacent point of \( P^j_k \) and are included in \( P_j \) which is the set of attachment points and contact-maintaining conditional contact points of \( m^j \) (Figure 3). We note that all elements of \( P_j \) are termed attachment points in the remainder of the paper for simplicity.

A muscle \( m^j \) exerts \( f_{at}^j \) between \( P_j \) on \( b^j_k \) and \( P_j \) on \( b^j_{k+1} \) through the path from \( P^j_k \) to \( P^j_{k+1} \). This is physically identical to the situation in which every pair of neighboring attachment points pulls each other with \( f_{at}^j \) because the magnitude of the tensile force is equal at any point on \( m^j \) and the direction of the tensile force changes.
through the path. The two force vectors $f_{j}^{k-}$ and $f_{j}^{k+}$ acting on $b_{j}^{k}$ at $p_{j}^{k}$ are:

$$f_{j}^{k-} = f_{int}^{0} \frac{p_{j}^{k-} - p_{j}^{k}}{|p_{j}^{k-} - p_{j}^{k}|}, \quad f_{j}^{k+} = f_{int}^{0} \frac{p_{j}^{k+1} - p_{j}^{k}}{|p_{j}^{k+1} - p_{j}^{k}|},$$ (7)

Here $f_{j}^{k-}$ and $f_{j}^{k+}$ are 0 because $p_{j}^{k}$ and $p_{j}^{k}$ are respectively the first and last attachment points of $m_{j}$.

If adjacent points $p_{j}^{k}$ and $p_{j}^{k+1}$ are attached to the same bone ($b_{j}^{k} = b_{j}^{k+1}$), $f_{j}^{k+}$ and $f_{j}^{k+1-}$ cancel each other out and do not have any effect on the entire body (see Figure 3 for an example). We exclude those points from $P_{j}$ to improve simulation performance.

### 3.3 Equation of Motion

The equation of motion of the humanoid model is written as:

$$M(q)\ddot{q} + c(q, \dot{q}) = J_{a}^{T} f_{a} + J_{c}^{T} f_{c},$$ (8)

where $q$, $\dot{q}$, and $\ddot{q}$ are the generalized position, velocity and acceleration of all DOFs, $M$ is the mass inertia matrix, and $c$ represents the Coriolis, centrifugal and gravitational force. The Jacobian matrices $J_{a}$ and $J_{c}$ map the generalized velocity $\dot{q}$ to the global velocities at the attachment points and ground contact points, respectively. The vector $f_{a}$ is the muscle forces at all attachment points which aggregates all $f_{i}^{k}(\cdot)$ of the model and $f_{c}$ is the ground contact force, which can be formulated as

$$f_{c} = V_{c} \lambda,$$ (9)

where $V_{c}$ is the linearized friction cone basis vectors, and $\lambda$ is a coefficient vector. The muscle forces $f_{a}$ can be expressed as:

$$f_{a} = V_{a} C f_{act},$$ (10)

where $V_{a}$ is unit direction vectors of all muscle forces $f_{i}^{k}(\cdot)$. $C$ is the converting matrix that relates the indices of muscles to the indices of attachment points, and $f_{act}$ is the aggregate vector of all scalar musculotendon forces $f_{i}^{0}$.

Substituting the contraction dynamics equation (5) into Equation (10) transforms the input activation level to the musculotendon force:

$$f_{a} = V_{a} C P (A a + p),$$ (11)

where $P$ is a diagonal matrix containing the cosines of the penna-angles, and $a$ is the muscle activations. $A$ is a diagonal matrix containing the active force scaling parameters $g_{ai}(l_{in})$ for all muscles. $p$ is a vector containing coefficients for the passive forces and passive damping elements $g_{pi}(l_{in})$ for all muscles. Given the assumption that our Hill-type muscle model is mass-less, the change of activation $a$ immediately affects the muscle fiber lengths $l_{in}$ and its time-derivatives $\dot{l}_{in}$, which in turn determines the musculotendon force $f_{act}$ according to the nonlinear muscle dynamics equations (Equation (1), (2), and (5)). $A$ and $p$ linearize this highly nonlinear relationship between the activation $a$ and force $f_{act}$ around the current values of $l_{in}$ and $\dot{l}_{in}$ such that they can be used in the quadratic program formulation of the muscle optimization. The final form of the equation of motion is:

$$M(q)\ddot{q} + c(q, \dot{q}) = J_{a}^{T} V_{a} C P (A a + p) + J_{c}^{T} V_{c} \lambda.$$ (12)

### 4 Muscle Optimization

The goal of the muscle optimization is to find the optimal coordination of muscle activation levels to control the musculoskeletal model. The muscle optimization is seamlessly integrated as a part of our controller and invoked in a per-frame basis. At every time steps at runtime, our many-muscle controller adjusts the reference motion using a balance strategy presented by Kwon and Hodgins [2010], which plans the balance-recovering reference motion instantaneously based on the estimated pendulum state, and then optimizes muscle actuations to track the adjusted reference motion.

#### 4.1 Objectives

The optimization at runtime uses four objectives to minimize efforts, contact forces, deviation from the reference motion, and deviation from the end-effector trajectories. All terms are instantaneous and formulated as a quadratic form with respect to optimization variables $\ddot{q}$, $a$, and $\lambda$.

**Effort.** Minimizing effort is important in solving the underdetermined muscle actuation problem in that the best muscle actuation among many possible coordinations can be found. The sum of squared activations measures instantaneous effort at any instance:

$$G_{e} = |a|^2.$$ (13)

**Contact Force.** Minimizing contact force reduces the impact from the ground and thus improves the stability of control.

$$G_{c} = |\lambda|^2.$$ (14)

**Tracking.** We compute the desired acceleration $\ddot{q}_{d}$ to track the balance-recovering reference motion as follows:

$$\ddot{q}_{d} = k_{d} f_{diff}(q_{r}, q) + k_{d}(q_{r} - \ddot{q}) + \ddot{q}_{r},$$ (15)

where $q_{r}$, $\dot{q}_{r}$, and $\ddot{q}_{r}$ are the reference position, velocity and acceleration of all DOFs, and $k_{d}$ and $k_{d}$ are the tracking gains. The function $f_{diff}$ computes the difference between two positional DOFs depending on the type of the corresponding joints. We use $k_{d} = 2\sqrt{K_{p}}$ for critical damping. The tracking objective minimizes the difference between $\ddot{q}_{d}$ and $\ddot{q}$:

$$G_{tr} = |\ddot{q}_{d} - \ddot{q}|^2.$$ (16)

**End Effectors.** Foot-step planning is essential for the biped to maintain its balance. The end-effector objectives reinforce the end-effectors to track their desired positional/angular trajectories more accurately. We apply the end-effector objectives for both feet and the torso.

$$G_{ee} = |y_{\ddot{d}} - y_{\ddot{r}}|^2,$$ (17)

where $y_{\ddot{d}}$ and $y_{\ddot{r}}$ are the desired and actual acceleration of the $i^{th}$ body part. The desired acceleration is linearly proportional to the position and velocity differences between the desired and the current end-effector configurations.

### 4.2 Constraints

We use one equality constraint, the equation of motion in (12), to make the simulation and control physically plausible. Inequality conditions are used for the Coulomb ground contact model and the
ranges of the muscle activations. The ground contacts are formulated as:

\[ \lambda \geq 0, \quad V^T_r \mathbf{J} \mathbf{q} + V^T_s \mathbf{J} \mathbf{q} + V^T_q \mathbf{J} \mathbf{q} \geq 0, \]

which respectively are the friction cone condition and the non-penetration, non-slipping condition [de Lasa et al. 2010]. The muscle activations are in the range of 0 and 1.

\[ 0 \leq a \leq 1. \]

The muscle activation level is zero when a muscle is fully relaxed exerting no active force, and one when a muscle is exerting its maximum active force.

4.3 Quadratic Programming Formulation

The muscle optimization step is formulated as a quadratic program using the objectives and linear constraints stated above:

\[
\begin{align*}
\text{minimize} & \quad g_{cl} G_{cl} + g_{ts} G_{ts} + \sum_i g_{ee} G_{ee}^i + g_{cf} G_{cf}, \\
\text{subject to} & \quad \text{Equation (12), (18), (19), and (20)},
\end{align*}
\]

where \( g_{cl}, g_{ts}, g_{ee} \) and \( g_{cf} \) are the weight constants for each objective and \( i \in \{ \text{left foot, right foot, torso} \} \).

5 Trajectory Optimization

The functionality of our trajectory optimization is twofold. As extensively discussed in previous work [Lee et al. 2010; Coros et al. 2010], modulating step locations is a key factor of maintaining balance in biped locomotion. The trajectory optimization modulates the reference motion (as well as its step locations) at the pre-processing phase such that our runtime controller can reproduce the original reference motion more accurately and robustly. It also allows us to change the reference motion more aggressively to adapt to new conditions and requirements. In the latter case, the original reference motion serves as a regularization term in the optimization process.

In the offline trajectory optimization, we optimize only the foot trajectories to adapt since they are the most essential components of full body gaits. Even though only the foot trajectories are optimized, the impact of the change often affects the runtime controller to make a full body change of the simulated motion. Each of the foot trajectories is represented by offsets of the swing and the stance foot position from the corresponding reference foot position at three uniformly distributed keyframes. Assuming the symmetry of the gait, the dimension of the search space is 18 (6 offset points in 3D space). The offsets are applied to the balance-recovering reference motion that is tracked in the muscle optimization.

The trajectory optimization uses five objective terms. The first two terms are mandatory involving essential functionalities (trajectory tracking and balancing) of the controller, while the others are optional terms that we can choose to specify new requirements for motion adaptation. All terms are designed to account for longer horizons of gait patterns.

Pose Difference. Being able to faithfully reproduce any given reference motion is a fundamental objective. We penalize the deviation of the simulated motion from the original reference motion.

\[ H_{pd} = \sum_{f} f_{pd}(\mathbf{q}_{f}, \mathbf{q}), \]

where \( N_{\text{fall}} \) is the number of simulation time slots before falling down and \( f_{pd}(\cdot) \) computes the pose difference by point cloud matching.

Falling Down. The humanoid model should maintain its balance not to fall over. This objective penalizes earlier falling down:

\[ H_{fd} = \sum_{1}^{N_{\text{fall}}} 0 + \sum_{N_{\text{fall}} + 1}^{N_{\text{final}}} 1, \]

where \( N_{\text{final}} \) is the total number of simulation time slots. If the model does not fall down during the simulation, \( H_{fd} \) is zero.

Efficiency. The metabolic energy consumption has often been employed to measure the effort of locomotion in literature. We found that minimizing energy consumption generally leads to slow walk with a shorter stride. Instead, we use the efficiency term that measures energy consumption per unit moving distance, similarly to those used by Anderson and Pandy [2001]:

\[ H_{ec} = -\frac{1}{D} \sum_{1}^{N_{\text{fall}}} \hat{E}, \]

where \( \hat{E} \) is the current rate of metabolic energy expenditure [Wang et al. 2012] and \( D \) is the total moving distance before falling down.

Contact Force. The contact force objective with a longer horizon is motivated by the pain-avoidance behavior. Minimizing ground contact force can reduce pain on injured joints.

\[ H_{cf}^i = \sum_{1}^{N_{\text{fall}}} \| \mathbf{f}_{ci}^i \|, \]

where \( \mathbf{f}_{ci}^i \) is the resultant contact force of the \( i^{th} \) foot and \( i \in \{ \text{left foot, right foot} \} \).

Muscle Force. We introduce the muscle force objective to simulate another form of pain-avoidance behavior. Minimization of the force of a specific muscle can be interpreted as the pain-avoidance behavior due to strain or injury. The objective is:

\[ H_{mf}^j = \sum_{1}^{N_{\text{fall}}} f_{mf}^j, \]

where \( f_{mf}^j \) is the musculotendon force of the \( j^{th} \) muscle.

The total objective function is:

\[ H = h_{pd} H_{pd} + h_{ee} H_{ee} + \sum_{i} h_{ee}^i H_{ee}^i + \sum_{j} h_{mf}^j H_{mf}^j, \]

where \( h_{pd}, h_{ee}, h_{ee}^i \) and \( h_{mf}^j \) are weight constants. To evaluate Equation (28), we run a simulation of 8 to 15 gait cycles using the runtime controller. The landscape of Equation (28) is highly nonlinear and even discontinuous, and thus difficult to optimize. To minimize the objective function, we use the Covariance Matrix Adaptation algorithm [Hansen 2006], which is a derivative-free, stochastic optimization technique.
Figure 4: Simulations of locomotion skills (top to bottom): normal walk with gait2592, leaning walk with fullbody, marching walk with fullbody, slow run with fullbody, in-place slow run with gait2592, in-place fast run with fullbody, and quick-step slow run with gait2562.

6 Results

We use our implementation based on the Lie group theory to build the equations of motion of the dynamic system (see [Park et al. 1995]). The quadratic program of the muscle optimization is solved at 120 Hz using Quadprog++. The simulation advances by integrating the results from the quadratic program. The simulation runs about eight to twelve times slower than real-time depending on the complexity of the humanoids. The muscle fiber lengths are updated at 840 Hz by using the contraction dynamics equation. The trajectory optimization for each example takes six to nine hours depending on the number of iterations (100 to 200) using 60 cores on a cluster of Intel Xeon E5-2680 machines. Muscles are rendered in blue when they are fully relaxed and in red when they are fully activated. The color is linearly interpolated for partial activation.

In the following examples, we always set the weights $h_{pd}$ and $h_{fd}$ in the trajectory optimization objective $H$ (in Equation (28)) to 0.1 and 10$^4$ and set $h_{ec}$, $h_{ct}$, and $h_{int}$ to zero unless otherwise specified. For the pain avoidance, muscle weakness and tightness, and joint dislocation examples, the controllers were optimized with the gait2592 model and normal walk as the reference motion.

Locomotion Skills. The first set of experiments was conducted to show that our scheme can reproduce captured reference motions. We optimized seven controllers named normal walk, leaning walk, marching walk, slow run, in-place slow run, in-place fast run, and quick-step slow run (Figure 4). Because the normal walk motion from the OpenSim distribution was without arm data, it is used only for gait2592 and gait2562. Other reference motions were from public data [Lee et al. 2010; SNU Motion Database 2013] and were optimized for all three humanoid models. All reference motions were retargeted to match the musculoskeletal models. When the reference motion was not long enough, we created a longer motion by stitching a few cycles together. For slow run, in-place slow run and in-place fast run, we scaled the isometric maximum forces of all muscles by two because the controller was not able to generate a stable cycle when the original maximum values from the OpenSim model file were used.

The trajectory optimization at the preprocessing phase is mandatory to produce stable cycles for all examples. It works well even with very energetic motions. Unlike previous work [Wang et al. 2010], we do not tune feedback gain parameters for each individual reference motion because the reference motions are robustly optimized for maintaining balance throughout the cyclic simulations given a reasonable set of control parameters.

Pain Avoidance. People having pain in a limb tend to show asymmetric gait disturbance. Our optimization scheme based on the detailed musculoskeletal models allows us to predict the gait patterns of people with muscle or joint pain.

First, we simulated a unilateral painful ankle plantar flexor. People with such problems tend to reduce the use of the ankle plantar flexor. We set the corresponding $h_{int}$ in Equation (28) to $10^2$. The total force of the affected ankle plantar flexor is 44% less than that of the contralateral one.

Second, we simulated walking locomotion with “arthralgia” or painful joints of a unilateral limb. Because the pain increases when the pressure inside each joint increases, we can simply assume that the pain increases as the contact force increases. The corresponding $h_{cf}$ in Equation (28) were set to 1. The optimized controller exhibits an antalgic gait pattern with a shorter stance duration of...
the affected limb. The total contact force and total stance duration of the affected limb are 55% and 70% less than those of the contralateral limb, respectively.

**Muscle Weakness, Tightness.** The next set of experiments was conducted while reducing the maximum muscle strength or increasing the tightness of specific muscles. The efficiency weight $h_{ec}$ in Equation (28) was set to 10 for these experiments.

We weakened the uni- or bilateral gluteus medii and gluteus minimi which act as abductors of the hip joints. For people who have weakness of bilateral gluteus medii, a “waddling” gait, a gait with an exaggerated lateral translation of the trunk, is observed. Our experiments reproduced the waddling gait with bilateral gluteus medii and gluteus minimi of which the maximum isometric force was scaled by 0.4. For people who have weakness of unilateral gluteus medius, a Trendelenburg gait, a gait with an exaggerated lateral translation of the trunk only in the direction of the weak muscles, is observed. Our experiments reproduced the Trendelenburg gait with those muscles of which the strength was scaled by 0.2. Our “many-muscle” control scheme allows us to simulate changes of muscles that control lateral movement, such as gluteus medius, which is not supported by the controller of Wang et. al. [2012].

We also weakened the uni- or bilateral ankle plantar flexors which play an important role in walking by generating a propulsion force when an ankle pushes off. Scaling a unilateral ankle plantar flexor by 0.1 resulted in a “limping” gait which does not depend on the propulsion force of the affected ankle. Scaling the maximum isometric force of the bilateral ankle plantar flexors by 0.2 produced a flexed knee gait.

Tightness of the hamstrings and psoas with weakness of the ankle plantar flexors is the most common reason for a “crouch” or a flexed knee gait in people who suffer from cerebral palsy. We increased the muscle tightness by shortening the tendon slack length which is the rest length of the tendon. By scaling the tendon slack length of the bilateral hamstrings and psoas by 0.8 and the maximum isometric force of the bilateral ankle plantar flexors by 0.2, our controller generates a crouch gait with a more flexed knee joint.

**Joint Dislocation.** In the case of neglected developmental dysplasia of the hip (DDH), the hip joint is dislocated in the superolateral direction. We simulated the gait with a unilateral DDH by moving the hip joint 3 cm in the lateral direction. The optimized gait showed a Trendelenburg gait because the fulcrum of the hip joint moved in the lateral direction, the center of mass of the entire body should move in that direction also, as observed in people with neglected DDH.

**Comparison with EMG data.** Figure 7 shows that our simulated activations match the timings and patterns of human EMG data. We compare simulated activations of our normal walk controller for the gait2592 with the EMG data of human walking reported by Demircan et al. [2009]. Although the EMG data is obtained from a different capture subject, it can provide general profiles of muscle activations during normal walking. The activation curves are low-pass filtered with a 10 Hz third order Butterworth filter which is similar to the filtering for generating EMG plots from raw data. The curves are averaged over all gait cycles. Note that the magnitude of surface EMG data are not reliable, and only their timings and patterns are relevant.

**External Pushes.** We tested the robustness of our controllers for external pushes. We applied force with a duration of 0.2 seconds at the joint between the pelvis and the trunk. From the right, left, rear, and front directions, our controller is resilient up to 80N pushes for normal walk and up to 160N pushes for slow run.

**Relative Intensity.** Our controllers can be used to investigate the relative intensities of various exercises by measuring the metabolic equivalent of task (MET) which is the rate of metabolic energy consumption normalized by body weight. The simulation results of the MET with gait2562 for locomotion skills are listed in Table 1. The simulated MET of the normal walk motion was 7.1.
This value was 2.4 times as large as that measured from an actual human [Ainsworth et al. 2011]. This discrepancy is probably due to the different energy metrics: the energy consumption of a real person is measured by analyzing the oxygen consumption, while the metabolic energy metric attempts to predict this based on quantities related to muscles. It is also possible that the error comes from the simplification of the effort metric in the muscle optimization. We would like to further investigate this problem in future research. In any case, the calculated energy consumption is still useful because one can obtain a rough idea about the relative intensity of an exercise.

7 Discussion

We proposed a novel formulation to resolve muscle redundancy, searching for the best actuations, accelerations and contact forces simultaneously. Muscle redundancy could also be resolved solely via static optimization, but we combine it with an online quadratic programming formulation which embeds muscle contraction dynamics to generate dynamic controllers. To the best of our knowledge, our work builds upon state-of-the-art technology of human locomotion control, with one of the most detailed public domain musculoskeletal models ever used by controllers with a widest variety of gaits ever demonstrated. Our controller is not sensitive to model parameters such as the number of muscles, skeletal structures or locomotion styles. It does not rely on any algorithm specifically designed for a given musculoskeletal model, such as the one used by Wang et al. [2012].

We intend to address important research problems in both graphics and biomechanics. In computer graphics, physics-based motion synthesis has been an important topic to gain physical realism of virtual humanoids. Especially, controller-based approaches produce simulated movements online, allowing humanoids to respond to unexpected disturbances. With hundreds of muscles whose parameters are from real-human data, our controller can accurately simulate the actuation process of human locomotion while adapting to changes in muscle parameters. This allows us to physically synthesize natural motions and express various locomotion styles simply by changing the optimization criteria or reference motions. On the other hand, many biomechanists have developed detailed musculoskeletal models, and reproduced human movements using trajectory optimization. Our work reformulates these approaches based on an online control algorithm. The controller approach allows us to predict movements while continuously adapting to new conditions. This has significant practical meaning for medical applications such as understanding and treating muscle disorders. For example, our simulator can be a framework for the virtual surgical planning of gait correcting surgery in patients with cerebral palsy.

Reference tracking plays a role of regularization the optimization in our framework, determining the shape of a solution when other terms do not affect the solution. The regularization term also provides a high-level guide to the solution, meaning that it indicates the key features of walking, such as take-off, swing, landing, and weight supporting. We set a small value for the pose difference weight not to make the tracking stiff. The tracking term does not restrict the solution space because we allow the reference motion to be altered during trajectory optimization.

Even though our muscle actuation model is highly realistic from the standard of dynamic gait simulation, there are still missing pieces, such as muscle activation dynamics, that have not been utilized yet. Muscle activation and deactivation dynamics are the processes that describe the delays (typically 5ms to 100ms) between the neural excitation arriving at the muscle and the muscle developing force. We did not incorporate activation/deactivation dynamics in our simulation because it did not make perceivable changes in the simulated result, while requiring smaller time steps and computational resource. Muscle actuation dynamics might be useful in future for understanding finer details of microscopic gait analysis.

A more accurate musculoskeletal model would be another interesting future work for the musculoskeletal controllers. For example, it would be interesting to apply time-varying aspects of real human muscles such as fatigues and injuries. Including such time varying properties would be useful for both animation and medical applications. More accurate modeling of the muscle geometries, volumetric deformations, and of intersections between them would generate a more appealing simulation of a human body. Controlling a larger repertoire of motor skills beyond locomotion would also be interesting and would extend our understanding of the mechanisms of human movement.

Acknowledgements

We thank to the reviewers for their helpful feedback. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2011-0018340) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (MSIP) (No. 2007-0056094).

Appendix A: Humanoid Models

We use three musculoskeletal models in the OpenSim file format (Figure 1). Each model description includes physical properties and relative locations of bodies and joints, and geometric and physiological properties of muscles such as the maximum isometric force. All models have polygonal mesh data for each bone. The vertices of each mesh are candidates for contact points during simulation.

Gait2592. This model is based on the gait2392_simbody model [Delp et al. 1990; Anderson and Pandy 1999] included in the OpenSim distribution. We make minor modifications to the OpenSim model for the easy of implementation. We merge each 1-DOF ankle joint and 1-DOF subtalar joint into one 3-DOF ankle joint because those two joints were very close to each other. The rotational center of the knee joints are assumed to be fixed in the local coordinates of the hip joints although the OpenSim model describes the knee-angle dependencies of the rotational center. The model has 25 DOFs and 92 muscles.

Gait2562. This model is based on the gait2354_simbody model, which is also included in the OpenSim distribution and has fewer muscles than gait2392_simbody. Along with the modifications on gait2592, two ankle evertors (peroneus longus on each leg) and six plantar/dorsi flexors (flexor hallucis longus, flexor digitorum longus and extensor digitorum longus on each leg) muscles are borrowed from gait2592_simbody, because gait2354_simbody has no ankle evertors and no muscles for the tib joints. The model has 25 DOFs and 62 muscles.

Fullbody model. We combine gait2562 and Dynamics Arms [Steele and Hammer 2013] to make a fullbody model. Similarly to modification made to the ankle joints, the radioulnar joints were merged into the wrist joints. The model has 39 DOFs and 120 muscles.

Appendix B: Contraction Dynamics

We use the contraction dynamics equations proposed by Thelen [2003] with some modifications. Note that forces and lengths used in following equations are normalized by the maximum isometric
force ($f_{m}^a$) and the optimal fiber length ($l_{m}^o$), respectively. The tilde symbols indicate normalized values.

**Force-Length Relationship of Tendon.** The tendon force-length relationship is defined as:

$$
\tilde{g}_t(\varepsilon_t) = \begin{cases} 
\frac{\tilde{F}_{t}^{toe}}{e^{\varepsilon_t \cdot \tilde{F}_{t}^{toe}} - 1} (e^{\varepsilon_t \cdot \tilde{F}_{t}^{toe}} - 1), & \varepsilon_t \leq \varepsilon_t^{toe} \\
\tilde{K}_{lin}(\varepsilon_t - \varepsilon_t^{toe}) + \tilde{F}_{t}^{toe}, & \varepsilon_t > \varepsilon_t^{toe}
\end{cases}
$$

(29)

where $\varepsilon_t$ is the tendon strain ($\varepsilon_t = (l - l_t^o)/l_t^o$ where $l_t^o$ is the tendon slack length), $\tilde{F}_{t}^{toe}$ and $\tilde{F}_{t}^{toe}$ define transition point of the curve from nonlinear to linear, $\tilde{K}_{lin}$ and $\tilde{K}_{t}$ are shape factors. We use $\tilde{F}_{t}^{toe} = 0.33$, $\tilde{K}_{t} = 3.0$, $\tilde{K}_{lin} = 1.712/\epsilon_t^e$, $\epsilon_t^{toe} = 0.6096\epsilon_t^e$ where $\epsilon_t^e$ is the tendon strain due to maximum isometric force [Thelen 2003]. Note that $g_t(l_t) = f_{m}^a \cdot \tilde{g}_t((l_t - l_t^o)/l_t^o)$.

**Passive Force-Length Relationship of Muscle.** The original passive muscle force-length relationship used by Thelen [2003] is defined as:

$$
\tilde{g}_p^o(l_m) = e^{\varepsilon_m^{opt}/\gamma - 1},
$$

(30)

where $l_m$ is the normalized muscle fiber length, $\varepsilon_m^{opt}$ is the passive muscle strain due to maximum isometric force and $\gamma$ is a shape factor.

However, Equation (30) generates small negative force when $l_m$ is smaller than 1, meaning that muscle fiber generates unrealistic “pushing” force.

We slightly modify the equation as follows:

$$
\tilde{g}_p(l_m) = \begin{cases} 
0, & l_m \leq 1 \\
\tilde{g}_p^o(l_m), & l_m > 1
\end{cases}
$$

(31)

Note that $g_p(l_m) = f_m^a \cdot \tilde{g}_p(l_m/l_m^o)$.

**Active Force-Length Relationship of Muscle.** The active muscle force-length relationship is defined as:

$$
\tilde{g}_a(l_m) = e^{-(l_m - 1)^2/\gamma},
$$

(32)

where $\gamma$ is a shape factor.

**Force-Velocity Relationship of Muscle.** The original inverse function of muscle force-velocity relationship used by Thelen [2003] is defined as:

$$
\tilde{g}_v^{-1}(a, \tilde{l}_m, \tilde{v}_{int}) = \tilde{l}_m^{max}(0.25 + 0.75a) \frac{\tilde{f}_{ce} - a\tilde{f}_1}{c},
$$

(33)

where $\tilde{l}_m$ is the normalized length of the musculotendon actuator and $f_1$ is $\tilde{g}_a(l_m)$. $\tilde{f}_{ce}$ is the normalized contractile element force of muscle fiber and $\tilde{f}_{ce} = \tilde{g}_a(\varepsilon_t)/\cos(\alpha) - \tilde{g}_p(l_m)$ by Equation (1), (2), and (5) without the passive damping term. Here $\tilde{l}_m^{max}$, $\tilde{l}_m^o$, and $\tilde{f}_{m}^{cen}$ are normalized maximum contraction velocity of muscle fiber and normalized maximum muscle force, and $A_t$ is a shape factor.

The force-velocity relationship $g_v$ might be a function of not only $\tilde{l}_m$, but $a$ or $\tilde{l}_m$ especially when $\tilde{l}_m$ is near $\tilde{l}_m^{max}$ [Zajac 1989]. By finding the inverse of Equation (33) analytically, we found $g_v$ used by Thelen is a function of $a$ and $\tilde{l}_m$. Because dependence of the force-velocity relationship on $a$ or $\tilde{l}_m$ is not significant due to relatively short duration of near-zero muscle force state [Zajac 1989], we decided to use modified version of Equation (33) of which the inverse is a function of only $\tilde{l}_m$:

$$
\tilde{g}_v^{-1}(a, \tilde{l}_m, \tilde{v}_{int}) = \tilde{l}_m^{max} \frac{\tilde{f}_{ce} - a\tilde{f}_1}{c},
$$

(35)

where $\tilde{f}_{ce} = \tilde{f}_{ce} - b \cdot \tilde{l}_m$ and $c$ can be calculated from Equation (34) by substituting $\tilde{f}_{ce}$ to $\tilde{f}_{ce}$. Because we employ the passive damping element, we use $\tilde{f}_{ce}$ instead of $\tilde{f}_{ce}$. Equation (35) is a quadratic equation in the variable $\tilde{l}_m$ and can be easily solved with the quadratic formula within a plausible range of $\tilde{l}_m$.

We can find the force-velocity relationship by inverting Equation (35) analytically:

$$
\tilde{g}_v(l_m) = \begin{cases} 
\tilde{l}_m^{max} + \frac{\tilde{l}_m^{max}}{\tilde{l}_m^{max} - \tilde{l}_m/A_t}, & \tilde{l}_m \leq 0 \\
\frac{\tilde{l}_m^{cen}(2 + 2/A_t) + \tilde{l}_m^{max}((\tilde{l}_m^{cen}) - 1)}{\tilde{l}_m(2 + 2/A_t) + \tilde{l}_m^{max}((\tilde{l}_m^{cen}) - 1)}, & \tilde{l}_m > 0
\end{cases}
$$

(36)

Note that $\tilde{f}_{ce} = a \cdot g_a(l_m) \cdot g_v(l_m) = a \cdot f_m^a \cdot \tilde{g}_a(l_m) \cdot \tilde{g}_v(l_m)$.

The muscle property $f_m^a$, $l_m^o$, $l_t^o$, $\varepsilon_{m}^{opt}$ are specified for each muscle in the humanoid model description. Our humanoids models use $\epsilon_t^{e} = 0.033$, $\epsilon_t^{opt} = 0.6$, $k_{ce} = 4.0$, $\gamma = 0.5$, $l^{max}_m = 10$, $f^{cen}_m = 1.8$, $A_t = 0.3$ for all muscles.

**Appendix C: Initial Muscle States**

The initial value of $a$ and $l_m$ at the start of simulation are not clear because they are invisible internal muscle states. We start the simulation with fully relaxed muscles, meaning that initial $a$ is zero. Then $l_m$ is computed to generate an isometric muscle force which is a force when $l_m$ is zero. We compute it by solving Equation (35) for $l_m$ given $a = 0$ and $l_m = 0$. Using Equation (1), (2), (5) instead of Equation (35) gives the same results.

**References**


