Flexible Muscle-Based Locomotion for Bipedal Creatures

Thomas Geijtenbeek*
Utrecht University

Michiel van de Panne
University of British Columbia

A. Frank van der Stappen
Utrecht University

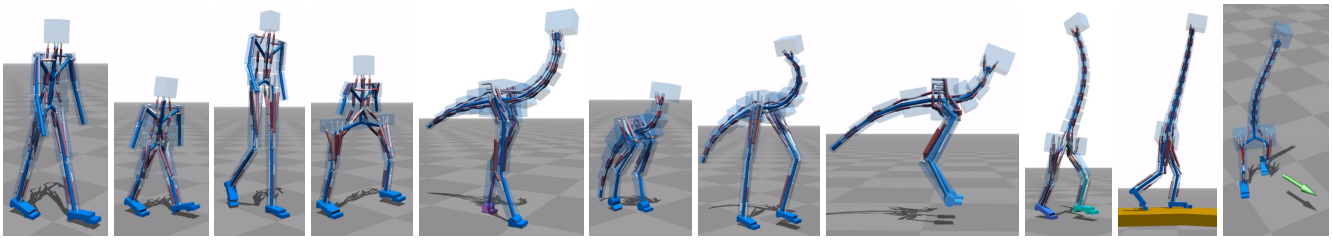


Figure 1: *Physics-based simulation of locomotion for a variety of creatures driven by 3D muscle-based control. The synthesized controllers can locomote in real time at a range of speeds, be steered to a target heading, and can traverse variable terrain.*

Abstract

We present a muscle-based control method for simulated bipeds in which both the muscle routing and control parameters are optimized. This yields a generic locomotion control method that supports a variety of bipedal creatures. All actuation forces are the result of 3D simulated muscles, and a model of neural delay is included for all feedback paths. As a result, our controllers generate torque patterns that incorporate biomechanical constraints. The synthesized controllers find different gaits based on target speed, can cope with uneven terrain and external perturbations, and can steer to target directions.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: physics-based animation, musculoskeletal simulation

Links: DL PDF

1 Introduction

Physics-based simulation is an established technique for the automatic generation of interactive natural-looking motion. To extend this approach to actively controlled virtual characters has been a longstanding research goal, in which tremendous progress has been made in recent years. Locomotion controllers have been developed that robustly deal with changes in character morphology, external perturbations and uneven terrain.

Unfortunately, in many cases the resulting motions are still not as natural as we would like. One common approach that can help improve the quality of the simulated motions is to use motion capture data as part of the control strategy. However, such methods are limited to characters and motions for which data is available. Furthermore, the biomechanical constraints that are implicit in captured motions are not preserved during the motion editing or motion retargeting that is often required to leverage limited motion data. Another approach for improving the motion quality has been to use optimization to help shape the motion, such as optimizing for minimal energy as well as task objectives. However, in the absence of biomechanical constraints, optimization objectives may lead to unnatural torque patterns or require cumbersome manual tuning. Commonly implemented joint limits and torque limits remain a crude approximation of the motion constraints that are implicit in articulated figures driven by musculotendon units.

More recently, emerging from biomechanics research, researchers have begun to develop methods that include biomechanical constraints into the simulation. Using such an approach, the natural gaits of various animals can be simulated without the need for motion data. However, the principal focus to date has been on modeling human motion, and the solutions remain limited in their locomotion abilities and robustness.

In this paper, we make the following contributions:

- We develop a control method and optimization strategy for the simulated locomotion of fully 3D bipedal characters, including imaginary creatures, that are driven entirely by simulated muscle-based actuation. The method produces robust locomotion at given speeds to target directions and does not require pre-existing motion data. The characters can further cope with modest variations in terrain.

- We introduce muscle routing optimization as an important feature that enables and simplifies the design of muscle-based control strategies for a variety of character morphologies. Instead of needing an exact musculoskeletal model, our method requires only an approximate template of where muscles are attached and routed. The specific geometry is then optimized within the specified ranges allowed by the template, along with the parameters related to the muscle control. This approach enables the discovery of efficient muscle routings for creature models for which there exist no real-world data to draw from.

- We make use of a muscle-based approximation to Jacobian transpose control as a core component of our framework. This enables a more creature-generic and motion-generic control architecture and is applied to the majority of joints in our creature models.

*e-mail:t.geijtenbeek@uu.nl

2 Related Work

Methods for physics-based character animation that use forward dynamic simulations have been a research focus for over two decades, most often with human locomotion as the motion of interest. A survey of the considerable body of work in this area can be found in [Geijtenbeek and Pronost 2012]. A rigid-link articulated figure is typically driven by treating the joint torques at each time step as free variables, constrained by joint angle limits and joint torque limits. The underlying balance strategies commonly make direct or indirect use of foot placement, e.g., [Raibert and Hodgins 1991; Hodgins et al. 1995; Laszlo et al. 1996; Yin et al. 2007; Tsai et al. 2010]. Basic locomotion capabilities have been extended in a variety of ways, including coping with terrain variations, character morphology variations, flexibly parameterized walking and running, new types of motions, new control abstractions, and methods for flexible motion sequencing. Examples here include [Faloutsos et al. 2001; Wang et al. 2009; Coros et al. 2009; Jain et al. 2009; Wang et al. 2010; Wu and Popovic 2010; Mordatch et al. 2010; de Lasa et al. 2010; Ye and Liu 2010; Coros et al. 2010; Jain and Liu 2011; Liu et al. 2012].

The use of motion capture data can greatly help in achieving natural physics-based locomotion. It can be used as a reference trajectory, as a well-chosen initialization point for an optimization, or as an example of an optimal solution from which to then generalize further solutions. A number of approaches exploit motion data to achieve torque-based control of simulated human motions [Liu et al. 2005; Sok et al. 2007; Yin et al. 2007; da Silva et al. 2008a; da Silva et al. 2008b; Muico et al. 2009; Lee et al. 2010; Kwon and Hodgins 2010; Jain and Liu 2011; Muico et al. 2011; Geijtenbeek et al. 2012]. Many of these methods further tackle aspects of parameterization, other classes of motion, and choice of feedback abstraction.

In reality, joint torques cannot be commanded at will and must instead arise from muscles that have their own activation dynamics, force production behavior, and moment arms that change over time. They also often do not provide direct control over individual degrees of freedom, as is assumed with computed torque methods. This is because a single muscle may span multiple joints, and a single joint may be spanned by multiple muscles. Biomechanics research has developed muscle-based approaches for the simulation of a variety of human and animal motions, including lizards [Ijspeert et al. 2007], cat hind limbs [Maufroy et al. 2008], human jumping [Pandy et al. 1992], human pedaling [Thelen et al. 2003], and human gait [Taga 1995; Anderson and Pandy 2001; Geyer and Herr 2010; Ackermann and van den Bogert 2012]. Relatedly, muscle-based simulations are being explored in the computer animation literature, where they are applied to modeling human hand motion [Tsang et al. 2005; Sueda et al. 2008], human upper body motion [Lee et al. 2010], and to evaluating the realism of human motion trajectories [Geijtenbeek et al. 2010]. Most notably, the work of Geyer and Herr [2010] has been used as the basis to animate a full 3D humanoid character by Wang et al. [2012]. The motion controls are optimized with respect to an objective function that combines metabolic energy consumption and several walking-task-specific terms related to head stability and torso orientation. Together with muscle reflex models, this then produces stable forward dynamics simulations of walking at a variety of speeds that are shown to closely match human walking data.

While the majority of work on physics-based character simulation is focused on modeling human motion, control strategies have also been developed to drive simulations of block-based creatures [Sims 1994], swimming creatures [Grzeszczuk and Terzopoulos 1995; Tan et al. 2011], walking birds or other fantastical bipeds [Coros et al. 2009; Coros et al. 2010; de Lasa et al. 2010], and quadrupeds [Coros et al. 2011]. Physically-plausible gaits are developed *de novo*, i.e., without motion capture data, by Wampler et al. [Wampler and Popović 2009; Wampler et al. 2013]. Alternate kinematic approaches are developed by Hecker et al. [2008] and Kry et al. [2009] for producing visually plausible gaits for arbitrary creature morphologies.

Our work is most closely related to the impressive state-of-the-art work by Wang et al. [2012]. We share many of the goals of this recent work, but with the following notable differences: (1) The models we develop use 3D muscles to drive all the motion of the entire body. In comparson, Wang et al. [2012] use planar muscles restricted to the sagittal-plane to control the lower body, and use classic torque-based methods to control the coronal lower-body motion and the entire upper body. Our models are entirely muscle driven. (2) Our framework optimizes for the best muscle routing geometry. As such, the user only needs to provide an approximate template for muscle insertions and attachments. This is of particular utility when designing imaginary creatures or disproportional humans, for which these parameters are not known. Proper routing of 3D muscles can greatly simplify control, and we show that this aspect of the optimization significantly contributes to the ability to synthesize plausible motions. (3) We optimize for muscle physiological properties, including rest length and maximum force. These are not known a priori when designing new creatures. (4) Our control system relies heavily on target features (positions and orientations of links), and uses a muscle-based Jacobian transpose approximation to help compute target muscle activations. This contributes towards a more generic control framework. In contrast, the feedback rules of Wang et al. [2012] are tailored around the human-specific reflex model developed by Geyer and Herr [2010]. (5) Our controllers are robust enough to traverse moderate terrain variations and can perform shallow turns. (6) Most significantly, our method can be applied to automatically achieve different gaits for a variety of creatures, including imaginary creatures.

We note that the above differences are not always entirely beneficial. Because our framework targets a wider variety of creatures with a control architecture that attempts to be more generic rather than being tailored to human models, our basic approach may not achieve the same human motion fidelity as the human-specific method and results presented by Wang et al. [2012]. However, researchers focusing on achieving a more faithful human gait (or any other gait) can easily extend our basic approach by adding domain-specific target features, objective terms or feedback rules.

3 Musculoskeletal Model

Our creature model consists of a hierarchy of rigid bodies, which are actuated using an established dynamic muscle model [Geyer and Herr 2010]. The biomechanical constraints incorporated in this model ensure the creation of realistic and smooth torque patterns. More specifically, it models the physiological properties of contractile muscle fibers and tendons (contraction dynamics), and the electro-chemical process that leads to changes in activation state (activation dynamics).

3.1 Muscle Contraction Dynamics

Muscles generate forces through muscle fibers contracting, based on the current activation state of the muscle. The dynamics of contraction is commonly modeled using a three element structure, also known as *Hill-type muscle model* (see Figure 2, top). It consists of the following elements:

- A *contractile element* (CE) that represents the muscle fibers that contract based on the muscle activation state.

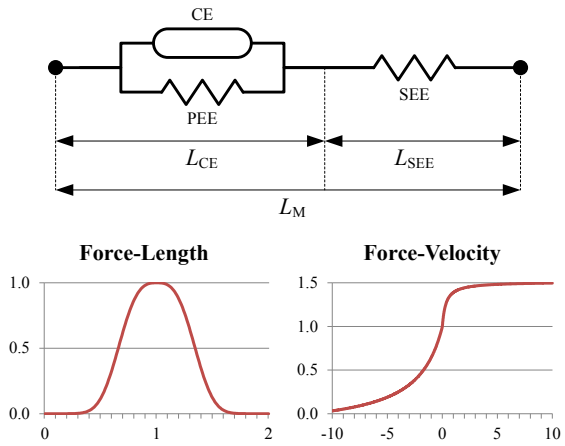


Figure 2: *Top: The three components of a Hill-type muscle. Bottom: Normalized force-length and force-velocity relations of the contractile element.*

- A *parallel elastic element* (PEE) that represents the passive elastic material surrounding muscle fibers.
- A *serial elastic element* (SEE) that represents the tendons that connect the muscle to the bones.

The force produced by the CE, F_{CE}, depends on the constant maximum isometrical force for the muscle, $\mathrm{F_{max}}$, the muscle activation a, fiber length L_{CE}, and contraction velocity V_{CE}:

$$F_{\mathrm{CE}} = a\,\mathrm{F_{max}}\,f_{\mathrm{L}}(L_{\mathrm{CE}})\,f_{\mathrm{V}}(V_{\mathrm{CE}}) \tag{1}$$

where f_{L} describes the relationship between force and length of a muscle, and f_{V} describes the relationship between force and the current contraction velocity (see Figure 2, bottom). Roughly speaking, a muscle can produce more force when its length is closer to its optimal length, and produces less force if it is contracting faster. The maximum isometric force $\mathrm{F_{max}}$ is a constant that we find through optimization.

The passive forces produced by the elastic elements, F_{PEE} and F_{SEE} are modeled as non-linear springs based on their length:

$$F_{\mathrm{SEE}} = f_{\mathrm{SEE}}(L_{\mathrm{M}} - L_{\mathrm{CE}}) \tag{2}$$
$$F_{\mathrm{PEE}} = f_{\mathrm{PEE}}(L_{\mathrm{CE}}) \tag{3}$$

where f_{SEE} and f_{PEE} are non-linear force-length relations and L_{M} represents the total muscle length, from which the length of the SEE can be derived. Analytical forms of f_{SEE}, f_{PEE}, f_{L} and f_{V} are described in [Geyer et al. 2003].

As the SEE is wired to the CE and PEE in series, the total muscle force F_{M} is subject to the force balance equation

$$F_{\mathrm{M}} = F_{\mathrm{CE}} + F_{\mathrm{PEE}} = F_{\mathrm{SEE}} \tag{4}$$

The length of the CE (from which the SEE length is derived) is initialized to be its optimal length, $\mathrm{L_{CE}^{opt}}$, which in combination with the tendon slack length $\mathrm{L_{SEE}^{slack}}$ defines the muscle rest length $\mathrm{L_{M}^{rest}}$:

$$\mathrm{L_{M}^{rest}} = \mathrm{L_{SEE}^{slack}} + \mathrm{L_{CE}^{opt}} \tag{5}$$

Both $\mathrm{L_{CE}^{opt}}$ and $\mathrm{L_{SEE}^{slack}}$ are important for the dynamic behavior of the muscle [Zajac 1989] and are subject to optimization. During simulation, activation state a and total muscle length L_{M} are input

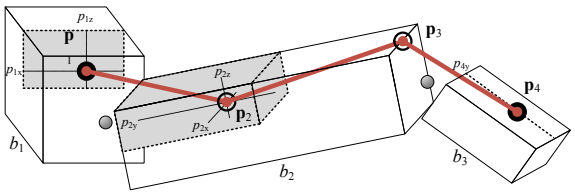


Figure 3: *Muscle attachment points that will be optimized within a constrained region. In this example, muscle point $\mathbf{p}_1$ is constrained to a 2D surface, muscle point $\mathbf{p}_2$ is constrained to a 3D volume, $\mathbf{p}_3$ is fixed, and $\mathbf{p}_4$ is constrained to a line. The actual areas used in our experiments are shown in Figure 8.*

parameters. The first is the result of activation dynamics (§3.2), the second is determined by the geometry of the current pose (§3.3). The muscle state parameter L_{CE} is updated through integration of the contraction velocity V_{CE}, which is derived from Equations (1) and (4), by inverting the force-velocity relation f_{V}:

$$\frac{\partial L_{\mathrm{CE}}}{\partial t} = f_{\mathrm{V}}^{-1}\left[\frac{F_{\mathrm{SEE}} - F_{\mathrm{PEE}}}{a F_{\mathrm{max}} f_L(L_{\mathrm{CE}})}\right] \tag{6}$$

For a detailed description of this procedure we refer to Geyer and Herr [2010].

3.2 Muscle Activation Dynamics

The activation state a of a muscle is altered as the result of a relatively slow electro-chemical process, based on a neural excitation signal u, which is output by the control system. This process is referred to as the *activation dynamics* and is modeled as:

$$\frac{\partial a}{\partial t} = \mathrm{c_a}(a - u) \tag{7}$$

in which $\mathrm{c_a}$ is the constant activation and deactivation rate. In our model we use $\mathrm{c_a} = 100\mathrm{s}^{-1}$, following Wang et al. [2012].

3.3 Muscle Geometry and Skeleton Interaction

The skeleton of a character and its muscles have a two-way interaction: muscles apply forces that change the skeleton pose, while the skeleton pose fully determines muscle length L_{M}, which then influences the contraction dynamics. The path of a muscle is determined by the locations where its tendons are attached to the bones, the bony landmarks around which the muscle wraps, and the time-varying poses of the joints that the muscle spans. In our model, we define a muscle path as a set of line segments connecting a fixed set of *via points*, which can be regarded as frictionless loopholes through which the muscle slides. This model is a simplification that has the advantage of high performance (which we require for our purpose) and omits the need to model skeleton geometry around which the muscles should wrap.

Muscle Geometry The routing of any muscle M is defined through a vector of n attachment points, $[\{b_1, \mathbf{p}_1\} \ldots, \{b_n, \mathbf{p}_n\}]$, each of which is defined by an offset point $\mathbf{p}_i$ and a body b_i to which the point is attached (see Figure 3). Each attachment point $\mathbf{p}_i$ is defined as a fixed offset in the coordinate frame of body b_i; it translates and rotates along with the body it is attached to. Multiple points can be attached to a single body. The first and last point, p_1 and p_n, represent the locations where the muscle tendons are attached to the skeleton, while the others are via points.

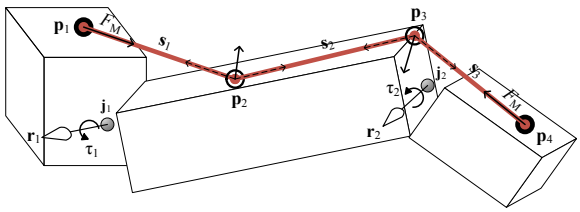


Figure 4: *Example muscle path with four attachment points, three bodies and two joints.*

The total muscle length L_{M} is equal to the summed lengths of the $n-1$ muscle segments, $[\mathbf{s}_1, \ldots, \mathbf{s}_{n-1}]$, which are found using the position of each point in the world coordinate frame, $\mathbf{p}_i^{\mathrm{W}}$:

$$L_{\mathrm{M}} = \sum_{i=1}^{n-1} \|\mathbf{s}_i\| \quad , \quad \mathbf{s}_i = \mathbf{p}_{i+1}^{\mathrm{W}} - \mathbf{p}_i^{\mathrm{W}} \tag{8}$$

The location of these attachment points has a great influence on both direction and magnitude of the torque a muscle can produce. The direction of the moment arm (and thereby its function) changes dynamically with the character pose; this relation is fully based on the locations of the attachment points. The amount of torque a muscle can provide depends on the projected distance between a muscle segment and the joint it spans. If it is further away, the moment arm is higher, but joint rotations will also lead to bigger changes in length, which limits the range in which the muscle can operate. Both aspects greatly affect control.

In our approach, we attempt to find efficient muscle routings through optimization. We do so by defining an area $\mathcal{P}_{\mathrm{i}}$ in which a muscle point $\mathbf{p}_{\mathrm{i}}$ must be contained. In our implementation, we allow variation of selected Cartesian components p_x, p_y or p_y within a range that is specified in an attachment template. Depending on the number of free components, the point $\mathbf{p}_{\mathrm{i}}$ is either fixed, or constrained to a line, a plane or a box (see Figure 3).

Force Application The total muscle length L_{M} is used in combination with the activation state a to compute the (scalar) muscle contraction force F_{M}. This force is transmitted to the skeleton at each attachment point to the body it is attached to, and generates a torque over each joint it spans. For each joint k, a torque τ_k is generated in the direction defined by moment arm $\mathbf{r}_k$. This moment arm corresponds to the cross-product between the direction of the muscle segment that crosses the joint, $\mathbf{s}_c$, and a vector from joint center $\mathbf{j}_k$ to any point on segment $\mathbf{s}_c$ (*e.g.* point $\mathbf{p}_c^{\mathrm{W}}$):

$$\tau_k = F_M \|\mathbf{r}_k\| \quad , \quad \mathbf{r}_k = (\mathbf{p}_c^{\mathrm{W}} - \mathbf{j}_k) \times \frac{\mathbf{s}_c}{\|\mathbf{s}_c\|} \tag{9}$$

The direction and size of moment arm $\mathbf{r}_k$ change as a function of the character pose, based on the geometry of $\mathbf{s}_c$ (see Figure 4).

4 Control

The goal of our muscle-based control system is to output muscle excitation signals that produce locomotion at a desired speed. An overview of our system is presented in Figure 5. At each step, we first update a finite state machine based on the current leg state. Next, we compose a set of target poses for a minimal set of featured body parts. These poses are based on a number of basic feedback rules for speed variation, heading control and balance. All parameters for constructing these poses are found through optimization. Finally, we compute the set of excitation signals that make muscle forces drive the featured body parts to their target positions and

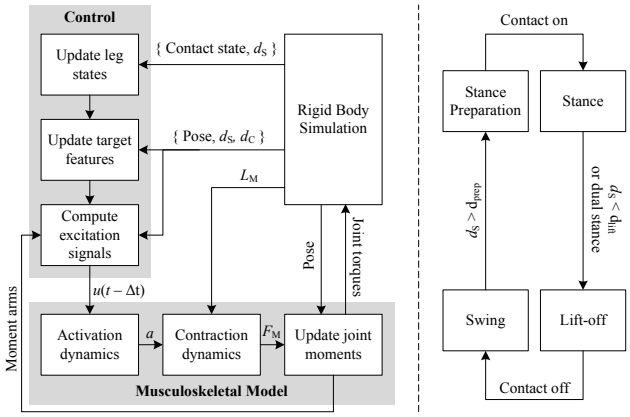


Figure 5: *System overview. At each simulation time step, the state of each leg is updated according to the finite state machine on the right (§4.1). After that, target features are computed (§4.2) from which desired muscle excitation signals are derived. The musculoskeletal model output (§3) is fed back to the physics simulation component.*

orientations using a muscle-based variation of Jacobian transpose control [Sunada et al. 1994]. We further use a small set of biologically inspired feedback rules. Time delay is incorporated in all of our excitation signals to simulate natural delay of neural systems.

4.1 Control States

For each leg, a separate finite state machine (FSM) keeps track of the current leg state. Similar to Wang et al. [2012], we distinguish between four states: *stance*, *lift-off*, *swing*, and *stance preparation* (see Figure 5). State transitions to stance and swing occur after ground contact changes. Ground contact is measured by comparing ground reaction force to a threshold value $\mathrm{F}_{\mathrm{contact}}$. Lift-off is initiated for the rear leg at dual stance, or when the signed horizontal distance between foot position and the center of mass, d_{S}, crosses a threshold value: $d_{\mathrm{S}} < \mathrm{d}_{\mathrm{lift}}$. Stance preparation is initiated after $d_{\mathrm{S}} > \mathrm{d}_{\mathrm{prep}}$. $\mathrm{F}_{\mathrm{contact}}$, $\mathrm{d}_{\mathrm{lift}}$ and $\mathrm{d}_{\mathrm{prep}}$ are subject to optimization.

4.2 Target Features

An important part of our control system is based on target poses, which we define for a set of featured body parts. More specifically, we define a target orientation for a *trunk* body, a target position and orientation for a *head* body, and target orientations for the *leg* segments during swing and stance preparation (see Figure 6). The legs in our model consist of at least three segments, named *upper*, *lower*, and *foot*. All target poses are defined in a world-aligned coordinate frame, specifically one that has axes aligned with the world 'up' axis and the current facing direction of the creature (defined by the orientation of the trunk segment). The parameters to construct these poses are found during optimization.

Trunk Target The target trunk orientation, $\tilde{\mathbf{Q}}_{\mathrm{trunk}}$ is composed of three angles, defined in the transversal, sagittal and coronal plane of the character (applied in that order). The transversal orientation is based on a target heading angle ψ_{heading}, which is a user input parameter. The sagittal orientation θ_{trunk} is based on the difference between the current center-of-mass velocity v_{COM} and desired forward velocity $\tilde{v}_{\mathrm{forward}}$, which is also a user input parameter. The adjustment of the sagittal orientation helps the character to lean forward and backward to accelerate or decelerate. The coronal orien-

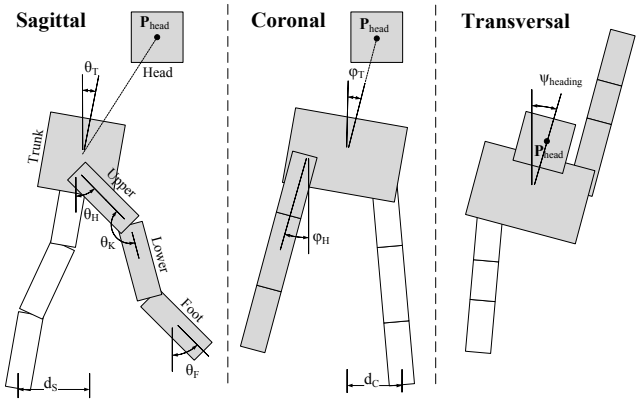


Figure 6: *Target features for trunk, head and swing leg, shown in sagittal, coronal and transversal projection.*

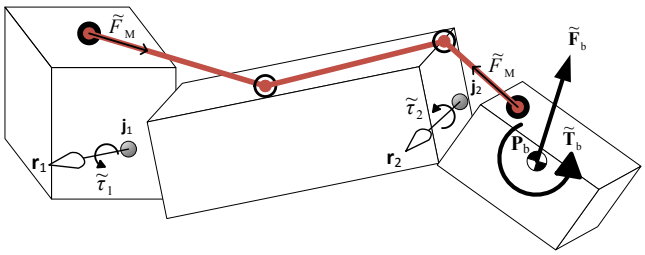


Figure 7: *Muscle-based feature control. A virtual force and torque are applied to the rightmost limb. For each muscle, the corresponding contraction force is computed based on the muscle moment arm(s). The excitation is computed based on the difference between desired and current muscle force.*

tation φ_{trunk} is based on the difference between current transversal trunk orientation ψ_{trunk} and target heading ψ_{heading}, and helps a character lean sideways towards its target heading:

$$\theta_{\text{trunk}} = \theta^0_{\text{trunk}} + \text{k}_\text{v}(\tilde{v}_{\text{forward}} - v_{\text{COM}}) \tag{10}$$

$$\varphi_{\text{trunk}} = \varphi^0_{\text{trunk}} + \text{k}_\text{h}(\psi_{\text{heading}} - \psi_{\text{trunk}}) \tag{11}$$

The target angular velocity $\tilde{\omega}_{\text{trunk}}$ is zero.

Head Target The head segment has both a target orientation and a target position. The target orientation $\tilde{\mathbf{Q}}_{\text{head}}$ is constant in the sagittal and coronal, and rotated to match ψ_{heading} in the transversal plane. Target position $\tilde{\mathbf{P}}_{\text{head}}$ is defined as a fixed offset in the trunk coordinate frame, and helps propagate the trunk orientation to the segments in the chain from trunk to head. The head also has a target linear velocity, $\tilde{\mathbf{v}}_{\text{head}}$, in the direction of ψ_{heading}, with a magnitude of $\tilde{v}_{\text{forward}}$. The target angular velocity $\tilde{\omega}_{\text{head}}$ is zero.

Leg Segment Targets The target upper leg orientation $\tilde{\mathbf{Q}}_{\text{upper}}$ is composed of three angles; in the transversal plane it is defined by target heading ψ_{heading}, while the orientation in the sagittal and coronal plane are based on fixed offsets and SIMBICON-style balance correction [Yin et al. 2007]:

$$\theta_{\text{hip}} = \theta^0_{\text{hip}} + \text{k}^\theta_\text{p}\, d_\text{S} + \text{k}^\theta_\text{d}(\tilde{v}_{\text{forward}} - v_\text{S}) \tag{12}$$

$$\varphi_{\text{hip}} = \varphi^0_{\text{hip}} + \text{k}^\phi_\text{p}\, d_\text{C} - \text{k}^\phi_\text{d} v_\text{C} \tag{13}$$

in which θ^0_{hip} and ϕ^0_{hip} are offset angles, while k^θ_p, k^θ_d, k^ϕ_p, and k^ϕ_d are control gains. The variables v_S and v_C are the current center-of-mass velocity in sagittal and coronal plane, while d_S and d_C are distances from the stance foot to the projected center-of-mass; see Figure 6. Initially, a rough estimate is provided for the offset angles (based on the initial pose of the character) and control parameters; their final values are found through optimization. For the sagittal parameters, different values are used during swing and stance preparation, leading to separate upper-leg targets for swing, $\tilde{\mathbf{Q}}^{\text{swing}}_{\text{upper}}$, and stance preparation, $\tilde{\mathbf{Q}}^{\text{prep}}_{\text{upper}}$. The upper leg has no target angular velocity during swing, while its target angular velocity during stance perparation, $\tilde{\omega}^{\text{prep}}_{\text{upper}}$, is zero.

The target orientation of the lower leg, $\tilde{\mathbf{Q}}_{\text{lower}}$, is defined through a fixed angle in the sagittal plane, θ_{knee}, relative to the upper leg orientation. The orientation of the foot segment, $\tilde{\mathbf{Q}}_{\text{foot}}$, is defined through a fixed angle in the sagittal plane, θ_{ankle}, relative to the ground plane. Both angles are initialized based on the model base pose, and then optimized. Both lower leg and foot have no target position or velocity.

4.3 Muscle-Based Feature Control

Given the set of target positions and orientations defined in the previous section, we wish to find the muscle excitation levels that cause any relevant muscle M to drive the featured bodies towards their respective targets. To accomplish this we use a muscle-based variation of Jacobian transpose control [Sunada et al. 1994], which attempts to find a set of muscle torques that emulate the effect of a virtual force or torque applied a specific body (see Figure 7). Specifically, for a body b with current state $\{\mathbf{P}_\text{b}, \mathbf{Q}_\text{b}, \mathbf{v}_\text{b}, \omega_\text{b}\}$ and target state $\{\tilde{\mathbf{P}}_\text{b}, \tilde{\mathbf{Q}}_\text{b}, \tilde{\mathbf{v}}_\text{b}, \tilde{\omega}_\text{b}\}$, we wish to minimize the state difference through a desired proportional-derivative (PD) feedback force $\tilde{\mathbf{F}}_\text{b}$ applied at $\mathbf{P}_\text{b}$, and a feedback torque $\tilde{\mathbf{T}}_\text{b}$:

$$\tilde{\mathbf{F}}_\text{b} = \text{k}_\text{P}\left[\tilde{\mathbf{P}}_\text{b} - \mathbf{P}_\text{b}\right] + \text{k}_\text{v}\left[\tilde{\mathbf{v}}_\text{b} - \mathbf{v}_\text{b}\right] \tag{14}$$

$$\tilde{\mathbf{T}}_\text{b} = \text{k}_\text{Q}\left[\mathbf{Q}_\text{b} \exp\left(\tilde{\mathbf{Q}}_\text{b}\mathbf{Q}_\text{b}^{-1}\right)\right] + \text{k}_\omega\left[\tilde{\omega}_\text{b} - \omega_\text{b}\right] \tag{15}$$

where $\exp(\tilde{\mathbf{Q}}_\text{b}\mathbf{Q}_\text{b}^{-1})$ is the 3D exponential map that represents the rotation from $\mathbf{Q}_\text{b}$ to $\tilde{\mathbf{Q}}_\text{b}$ (both of which are 3×3 rotation matrices). We attempt to emulate the effect of $\tilde{\mathbf{F}}_\text{b}$ and $\tilde{\mathbf{T}}_\text{b}$ through a set of torques in each joint k that is part of a chain from target body b, to a root body that is assumed stable. Each muscle M that spans over joint k produces a torque in the degree of freedom defined by moment arm $\mathbf{r}_\text{k}$; see Equation (9). The magnitude of this torque should match the effect of applying $\tilde{\mathbf{F}}_\text{b}$ and $\tilde{\mathbf{T}}_\text{b}$, thereby minimizing the difference between current and desired state for body b. To find this desired torque, $\tilde{\tau}_\text{k}$, we regard the rate of change of $\mathbf{P}_\text{b}$ and $\mathbf{Q}_\text{b}$, given a rotation α_k about the axis of moment arm $\mathbf{r}_\text{k}$:

$$\tilde{\tau}_\text{k} = \frac{\partial \mathbf{P}_\text{b}}{\partial \alpha_k}^T \tilde{\mathbf{F}}_\text{b} + \frac{\partial \exp\left(\mathbf{Q}_\text{b}\right)}{\partial \alpha_k}^T \tilde{\mathbf{T}}_\text{b} \tag{16}$$

$$= \left[\frac{\mathbf{r}'_\text{k}}{\|\mathbf{r}_\text{k}\|} \times \left[\mathbf{P}_\text{b} - \mathbf{j}_\text{k}\right]\right]^T \tilde{\mathbf{F}}_\text{b} + \frac{\mathbf{r}_\text{k}}{\|\mathbf{r}_\text{k}\|}^T \tilde{\mathbf{T}}_\text{b} \tag{17}$$

Based on Equation (9), we can derive the desired (scalar) muscle force $\tilde{F}_\text{M}$, using the magnitude of moment arm $\mathbf{r}_\text{k}$, and averaging for all m joints over which M spans:

$$\tilde{F}_\text{M} = \sum_{k=1}^{m} \frac{\tilde{\tau}_\text{k}}{m\|\mathbf{r}_\text{k}\|} \tag{18}$$

Note that the effect of a muscle force $\tilde{F}_\text{M}$ only approximates the desired joint torques. The final force and torque applied to body b will

in practice deviate from $\tilde{\mathbf{F}}_{\mathrm{b}}$ and $\tilde{\mathbf{T}}_{\mathrm{b}}$, and depend on the geometry and state of the individual muscles included in the chain of bodies, as well as the stability of the root body. However, we found this approximation to work well in practice, especially since the gains for $\tilde{\mathbf{F}}_{\mathrm{b}}$ and $\tilde{\mathbf{T}}_{\mathrm{b}}$ are subject to optimization.

The desired activation level $\tilde{a}_{\mathrm{M}}$ is estimated based on the maximum isometric force $\mathrm{F}_{\max}$:

$$\tilde{a}_{\mathrm{M}} = \frac{\tilde{F}_{\mathrm{M}}}{\mathrm{F}_{\max}} \tag{19}$$

In this estimation, we ignore the current length and velocity that are part of Equation (1). We have found that including length and velocity relations into our approximate inverse model for the contraction dynamics causes oscillations in muscle activation, because length and velocity change significantly over the course of neural and activation delay.

4.4 Muscle Activations

To compute the muscle excitation levels for our control system, we use a combination of muscle-based feature control, positive force-feedback [Geyer et al. 2006], and constant excitation values. We omit the use of length-based feedback rules defined in Geyer and Herr [2010] and Wang et al. [2012] in favor of our muscle-based feature control. A time delay is added to all feedback paths to simulate neural delay. For each muscle M involved in muscle-based feature control, we set the output excitation to match the desired activation:

$$\tilde{u}_{\mathrm{M}} = \tilde{a}_{\mathrm{M}}{}^{t-\Delta \mathrm{t}} \tag{20}$$

in which $t-\Delta\mathrm{t}$ represents the application of a delay $\Delta\mathrm{t}$. Following Geyer and Herr [2010], we use $\Delta\mathrm{t} = 20$ms for muscles attached to the foot, $\Delta\mathrm{t} = 10$ms for muscles attached to lower leg, and $\Delta\mathrm{t} = 5$ms for muscles attached to upper leg. For other muscles, we use $\Delta\mathrm{t} = 5$ms. Alternatively, the amount of delay could directly be derived from the distance of the muscle to the brain.

Trunk Orientation Feedback During stance, the orientation of the trunk is stabilized and rotated towards its target orientation $\tilde{\mathbf{Q}}_{\mathrm{trunk}}$ by all muscles connecting the trunk segment to a stance leg segment. The excitation for each HIP muscle corresponds to:

$$u_{\mathrm{HIP}}^{\mathrm{stance}} = \tilde{a}_{\mathrm{HIP}}(\tilde{\mathbf{Q}}_{\mathrm{trunk}}, \tilde{\omega}_{\mathrm{trunk}})^{t-\Delta \mathrm{t}} \tag{21}$$

Note that this feedback performs a rotation in 3 dimensions, depending on the planes in which the muscles operate.

Head Position and Orientation Feedback The head is moved towards its target state by all muscles connected to any body in the chain from trunk to head. The excitation is defined as:

$$u_{\mathrm{S}} = \tilde{a}_{\mathrm{S}}(\tilde{\mathbf{Q}}_{\mathrm{head}}, \tilde{\omega}_{\mathrm{head}}, \tilde{\mathbf{P}}_{\mathrm{head}}, \tilde{\mathbf{v}}_{\mathrm{head}})^{t-\Delta \mathrm{t}} \tag{22}$$

Stance and Lift-Off Feedback For the stance leg, we do not define target orientations or positions. Instead, we rely on positive force feedback to achieve natural joint compliance [Geyer et al. 2006]:

$$u_{\mathrm{F}+} = \mathrm{k}_{\mathrm{M}}^{\mathrm{F}+} F_{\mathrm{M}}{}^{t-\Delta \mathrm{t}} \tag{23}$$

in which $\mathrm{k}_{\mathrm{M}}^{\mathrm{F}+}$ is a constant feedback gain found during optimization. The length-force and velocity-force relations of any muscle ensure the excitation level does not increase indefinitely. Positive force feedback is applied to any muscle that extends the knee or ankle joint during stance and lift-off.

Subject	Parameters	Section
Muscle physiology	3–30 *	3.1
Muscle geometry	12–39 *	3.3
State transition	3	4.1
Target features	14	4.2
Feedback control	14–63 *	4.3, 4.4
Initial character state	6	†

Table 1: *Parameters subject to optimization. The number of parameters marked * is model dependent (see Table 3). † The parameters for initial character state are: initial forward lean, and initial speeds for upper swing leg, lower swing leg (and foot), upper stance leg, lower stance leg (and foot), and other bodies.*

During lift-off, all muscles attached to the hip are fed a constant excitation of high magnitude, to initiate a leg swing. The sign of this constant depends on whether the muscle is anterior (in front), or posterior (in the back). In addition, any knee extensor muscle is fed a constant negative excitation to initiate knee swing velocity:

$$u_{\mathrm{HIP}}^{\mathrm{lift}} = \mathrm{c}_{\mathrm{HIP}}^{\mathrm{lift}} \tag{24}$$

$$u_{\mathrm{KNEE}}^{\mathrm{lift}} = \mathrm{c}_{\mathrm{KNEE}}^{\mathrm{lift}} \tag{25}$$

Swing and Stance Preparation Feedback The upper leg is guided towards its target orientation during swing and stance preparation. The upper-leg target orientation and control use separate parameters for swing and stance preparation. The lower leg is guided towards its target only during stance preparation, while the knee remains passive during swing. The ankle muscles guide the foot towards its target orientation during both swing and stance preparation. The full set of feedback rules is as follows:

$$u_{\mathrm{HIP}}^{\mathrm{swing}} = \tilde{a}_{\mathrm{HIP}}(\tilde{\mathbf{Q}}_{\mathrm{upper}}^{\mathrm{swing}})^{t-\Delta \mathrm{t}} \tag{26}$$

$$u_{\mathrm{HIP}}^{\mathrm{prep}} = \tilde{a}_{\mathrm{HIP}}(\tilde{\mathbf{Q}}_{\mathrm{upper}}^{\mathrm{prep}}, \tilde{\omega}_{\mathrm{upper}}^{\mathrm{prep}})^{t-\Delta \mathrm{t}} \tag{27}$$

$$u_{\mathrm{KNEE}}^{\mathrm{prep}} = \tilde{a}_{\mathrm{KNEE}}(\tilde{\mathbf{Q}}_{\mathrm{lower}})^{t-\Delta \mathrm{t}} \tag{28}$$

$$u_{\mathrm{ANK}}^{\mathrm{swing}} = u_{\mathrm{ANK}}^{\mathrm{prep}} = \tilde{a}_{\mathrm{ANK}}(\tilde{\mathbf{Q}}_{\mathrm{foot}})^{t-\Delta \mathrm{t}} \tag{29}$$

Constant Excitation In addition to the feedback rules stated above, all muscles have a constant excitation, which is defined separately for {stance, lift-off} and {swing, stance preparation}:

$$u_{\mathrm{M}}^{\mathrm{stance,lift}} = \mathrm{c}_{\mathrm{M}}^{\mathrm{stance,lift}} \tag{30}$$

$$u_{\mathrm{M}}^{\mathrm{swing,prep}} = \mathrm{c}_{\mathrm{M}}^{\mathrm{swing,prep}} \tag{31}$$

5 Optimization

Both our muscle model (Section 3) and control model (Section 4) introduce a large number of free parameters, which are determined through off-line optimization (see Table 1 for an overview). The total set of parameters, $\mathbf{K}$, is optimized using Covariance Matrix Adaptation [Hansen 2006], with step size $\sigma = 1$ and population size $\lambda = 20$.

Objective The goal of our optimization process is to minimize the error $\bar{E}(\mathbf{K})$, which consists of the following components:

$$\bar{E}(\mathbf{K}) = \bar{E}_{\mathrm{speed}} + \bar{E}_{\mathrm{headori}} + \bar{E}_{\mathrm{headvel}} + \bar{E}_{\mathrm{slide}} + \bar{E}_{\mathrm{effort}} \tag{32}$$

	$E^{\text{vel}}_{\text{speed}}$	$E^{\text{ori}}_{\text{head}}$	$E^{\text{vel}}_{\text{head}}$	E_{slide}	E_{effort}
Weight ($\mathrm{W_m}$)	100	10	10	10	0.1
Treshold ($\mathrm{H_m}$)	0.1	0.2	0.3	0.2	0

Table 2: *Weights and thresholds for the individual error measures.*

Each right hand term is acquired by integrating a time dependent measure $E_{\mathrm{m}}(t)$ over a specific duration $\mathrm{t_{max}}$:

$$\bar{E}_{\mathrm{m}} = \mathrm{W_m} \left\{ \int_0^{\mathrm{t_{max}}} E_{\mathrm{m}}(t)\, \partial t \right\}_{\mathrm{H_m}} \tag{33}$$

in which $\mathrm{W_m}$ is measure-specific weight, while $\{\}_{\mathrm{H_m}}$ enforces a measure-specific threshold: the value between braces is set to zero if it is lower than $\mathrm{H_m}$. This allows for a prioritized optimization, as heavily weighted terms have greater influence until they reach their threshold. A significant difference with the error measure of Wang et al. [2012] is that we apply this threshold *after* integration, allowing incidental high values to be compensated by below-threshold averages. This is especially relevant for the initial stage of the simulation, when a character is still finding its pace. The individual weights and thresholds for each of the error terms are shown in Table 2.

The measure for target speed, $E_{\mathrm{speed}}(t)$, is computed as the normalized difference between base speed and target velocity $\tilde{v}_{\mathrm{forward}}(t)$:

$$E_{\mathrm{speed}}(t) = \| 1 - \frac{v_{base}(t)}{\tilde{v}_{\mathrm{forward}}(t)} \| \tag{34}$$

in which $v_{base}(t)$ is the forward speed based on the average foot position, updated at each contact initiation. To increase head stability, we use an error measure $E_{\mathrm{headori}}(t)$ for deviation of head orientation from its target, and $E_{\mathrm{headvel}}(t)$ for deviation for linear head velocity from its target:

$$E_{\mathrm{headori}}(t) = \| \mathbf{Q}^{-1}_{\mathrm{head}}(t) \tilde{\mathbf{Q}}_{\mathrm{head}}(t) \| \tag{35}$$

$$E_{\mathrm{headvel}}(t) = \| \tilde{\mathbf{v}}_{\mathrm{head}} - \mathbf{v}_{\mathrm{head}}(t) \| \tag{36}$$

In some of our simulations, we experienced a local minimum as the result of foot sliding. Error measure $E_{\mathrm{slide}}(t)$ prevents this by penalizing through average contact velocity $v_{\mathrm{contact}}(t)$:

$$E_{\mathrm{slide}}(t) = v_{\mathrm{contact}}(t) \tag{37}$$

For effort minimization $E_{\mathrm{effort}}(t)$, we use the current rate of metabolic expenditure [Wang et al. 2012].

Termination Conditions During the evaluation of $E(\mathbf{K})$, we terminate a simulation prematurely when failure is detected to save on simulation time and to help prevent local minima in the optimization. The following tests are performed during each time step:

- *Center-of-Mass Height*. To detect falling, we measure the center-of-mass position and compare its height to the initial state. The simulation is terminated if the measured height falls below a certain threshold. We use a factor of 0.9.
- *Heading*. We compare the target heading ψ_{heading} to the current trunk heading ψ_{trunk}, and terminate if they deviate over 45 degrees. In addition to keeping the character from drifting, this helps avoid a local minimum scenario where a character thrusts its feet forward during a backwards turn.
- *Self-Collision and Leg-Crossing*. We terminate on both self-collision and leg crossing, to avoid local minima where a character is unable to take another step because of self-collision. Leg-crossing occurs when the coronal left and right foot positions are reversed.

	Property	Human Legs	Human Body	Inverted Legs	Neck-Tail Body
Elements	Bodies	9	8	9	N + T + 2
	Joint DOFs	10	17	10	3N + 3T + 4
	Total muscles	20	20	20	4N + 4T + 4
	Unique muscles	10	6	10	4
Parameters	Muscle Property	30	6	30	3
	Muscle Topology	37	26	39	12
	Active Control	60	20	60	14
	Passive Control	3	1	3	0
	Total	130	53	132	29

Table 3: *Overview of structure and parameters of our model templates.*

After termination, we set $E_{\mathrm{speed}}(t) = 1$ for the remaining duration, resulting in a large penalty for failure. For the other measures we use $E_{\mathrm{m}}(t) = 0$ for the remaining duration, to ensure their effect is minimal during the early stage of development, in which a controller is only able to take a few steps.

6 Experiments

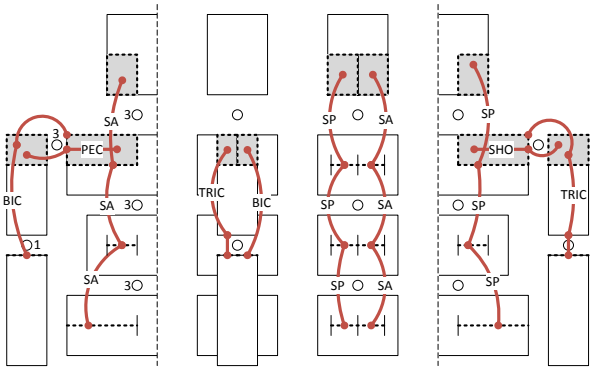
We test our framework using Open Dynamics Engine (ODE) [Smith 2006], version 0.8.2, using an integration time step of 0.0003s. We use the same ground contact model as Wang et al. [2009], simulating a spring-damper system with $\mathrm{k_p} = 75000$ and $\mathrm{k_d} = 2000$ through use of ODE's CFM and ERP parameters (see [Smith 2006] for details), and using a friction coefficient of $\mu = 10$.

We initialize the activation level of all muscles to 0.02, and force the left leg into swing state by rotating the upper leg 20 degrees forward. Depending on the type of experiment, we evaluate for a $\mathrm{t_{max}}$ of either 10s or 20s. The total optimization time depends on the character model and the type of experiment; the number of evaluated generations varies between 500 and 3000. On a standard PC, optimization time takes between 2 and 12 hours. For some experiments, we use the results of earlier optimizations to speed up optimization. Aside from those cases, all parameters were initialized using a single set of values for all models and target speeds. Once optimized, a controller can drive the simulated motion in real time.

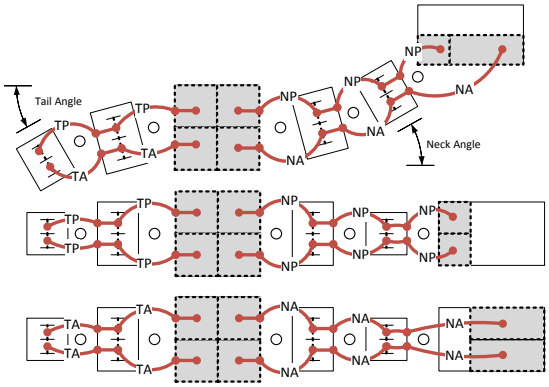
6.1 Creature Models

We have tested our method with four different model templates, which can be combined and parameterized to obtain a variety of biped characters as shown in Figure 1 and the supplementary video material. The free areas for muscle attachments, as well as the body hierarchy of each of these models is illustrated in Figure 8. The elements and free parameters for each of our models are summarized in Table 3.

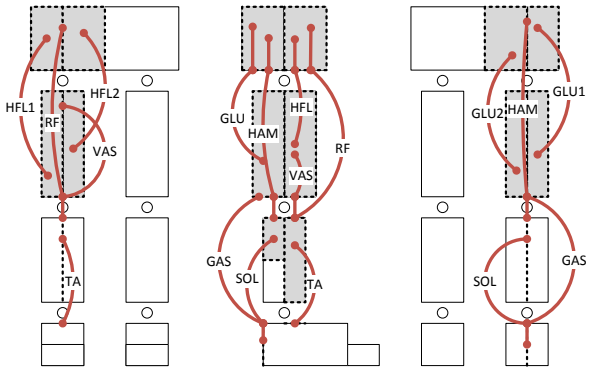
Humanoid Lower-Body Model The muscles in our humanoid lower-body model (Figure 8(c)) largely uses the same muscles as the sagittal-plane lower-body model of Wang et al. [2012], with the exception of the hip muscles. We use two out-of-plane hip flexors and two hip abductors, each of which operates simultaneously in the sagittal and coronal plane. The model has a total of 10 unique muscles for each leg; for each we optimize $\mathrm{F_{max}}$, $\mathrm{L^{opt}_{CE}}$ and $\mathrm{L^{slack}_{SEE}}$. The hip and ankle joint have 3 DOFs, while the knee has 1 DOF. We include passive spring-dampers for axial hip, axial ankle and



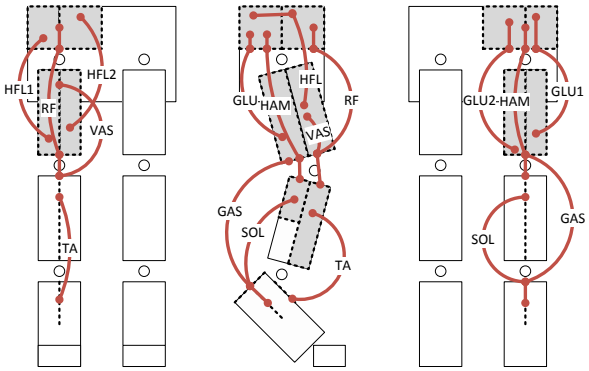
(a) Humanoid upper-body model: front, side arm, side body, and back.



(b) Neck-Tail upper-body model: side, top and bottom. The tilt angles and number of neck / tail segments can be adjusted.



(c) Humanoid lower-body model: front, side and back.



(d) Inverted lower-body model: front, side and back.

Figure 8: *Schematic overview of the permitted areas for the attachment points of each individual muscle. During optimization, each attachment point (the red dots) is allowed to move along the dotted line or area in which it is defined.*

planar ankle rotation; the gains are subject to optimization. Following Wang et al. [2012], we attach a toe segment to the foot using a joint with a spring constant of 30 Nm / rad.

Humanoid Upper-Body Model The humanoid upper-body model (Figure 8(a)) contains 2 upper-body segments (in addition to the shared root segment), a head segment and two arm segments. The elbow joint has 1 DOF, while the shoulder, spine and neck joints have 3 DOFs. There is a total of 12 spine muscles (6 on each side) that control head position and orientation, but properties are shared for all anterior (SA) and all posterior (SP) spine muscles. The arms are controlled using 4 unique muscles per arm that roughly represent various muscle groups present in the human body. These muscles are controlled through state-dependent constant excitation only; the arm states are linked to the corresponding legs. In the upper body muscles, we optimize L_{CE}^{opt} and derive L_{SEE}^{slack} from that, because allowing rest-length optimization for spine muscles often results in stiff short muscles during optimization. We add passive spring-damper control (with optimized gains) for spine joints in the axial direction.

Inverted Lower-Body Model The inverted lower-body model (Figure 8(d)) is similar to the humanoid lower-body, with the 'inverted knee' corresponding to the humanoid ankle. The template for muscle attachment has been modified to support long tilted feet and short upper legs, with hip joints located within the trunk body. Despite the differences in function, the initial values of the control parameters are the same as the humanoid lower-body leg model.

Neck-Tail Upper-Body Model The neck-tail upper-body model (Figure 8(b)) supports upper bodies with a variable number of neck (N) and tail (T) segments, as well as user-defined tilt angles. Each segment is attached to its parent with a 3 DOF joint and 4 muscles. There are 4 unique muscle types: neck anterior (NA), neck posterior (NP), tail anterior (TA), and tail posterior (TP); all muscles of the same type share the same control parameters and muscle properties. The neck muscles control the target head position and orientation through muscle-based feature control; see Equation (22). Note that while there are individual muscles to control the motion of each neck joint, their actions are implicitly coordinated via the feature-based control strategy. The tail muscles are controlled only through constant excitation, depending on the leg state of the corresponding side. All neck and tail joints contain a low-force critically damped spring, with a spring constant of 5 Nm / rad for all DOFs.

6.2 Controller Capabilities

The previously described lower and upper body models can be combined by sharing the root body. Further variation can be accomplished by changing character dimensions and (for the neck-tail model) number of segments and tilt angles. We have tested a number of capabilities of our controller, for the following combinations:

- *Human*. Humanoid lower-body model with average leg lengths (upper = 0.4, lower = 0.4, foot = 0.21), and average-

Model	Speed	Gen	Slope	Push	Turning
Human	2.0m/s	197	±5 deg	100N	12 deg /s
Human	4.0m/s	760	±3 deg	50N	12 deg /s
Ostrich	1.2m/s	698	±5 deg	50N	6 deg /s
LongNeck	1.5m/s	541	±5 deg	100N	12 deg /s
LongLegs	2.0m/s	403	±3 deg	100N	12 deg /s

Table 4: *Results for maximum slope, pushing force and turning speed for a selection of character models.* **Gen** *represents the number of generations before a straight walking controller reaches all of its objective function thresholds.*

sized humanoid upper-body model.

- *Ostrich*. Inverted lower-body model (upper = 0.2, lower = foot = 0.45), and neck-tail upper-body model with 5 neck and 5 tail segments.
- *LongNeck*. Humanoid lower-body model with relatively short legs (upper = lower = 0.3m, foot = 0.21) and a torso of 0.5m, and neck-tail upper-body model with 8 neck segments and no tail.
- *LongLegs*. Humanoid lower-body model with long legs (upper = lower = 0.6m), same upper-body model as *Ostrich*.

Tests include locomotion at multiple speeds, steering towards target headings, robustness over uneven terrain, and robustness in the face of external perturbations. The supplementary video material displays many of these capabilities. For robustness and steering tests, we initialize controllers with results from straight walking optimizations, and declare success when a stable controller is found within 300 generations (using $t_{max} = 20$).

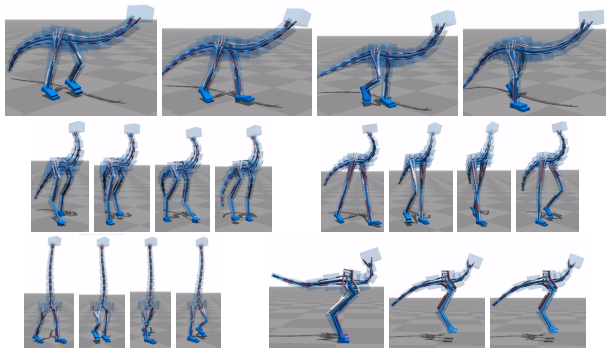


Figure 9: *Example synthesized walking and hopping gaits.*

Speed variation Similar to Wang et al. [2012], we automatically acquire different gaits by only varying the optimization target speed. In addition to seeing running gaits at higher speeds for humanoid models, we witness a hopping style gait emerging for some creatures with inverted lower-body models, only by increasing the target velocity.

For a selection of models, Table 4 shows the number of generations during optimization before all error measures in the objective function reach their thresholds.

Steering In this experiment we show the ability of our controllers to make interactive turns through variation of target heading ψ_{heading}. We do so by training controllers to follow a path with random turns, with a fixed turning velocity. The maximum turning velocities for a selection of creature models are shown in Table 4.

Uneven Terrain We examine the ability of our controllers to cope with uneven terrains by placing the creature models on a 1m wide ramp, with a steepness that randomly varies each meter. Table 4 shows a selection of maximum steepness ranges.

External Perturbations We examine the capability of our controllers to withstand external perturbations by applying pushes in random directions with a duration of 0.2s to the trunk body (or upper torso in case of humanoid upper-body models). Results are displayed in Table 4. Compared to Wang et al. [2012], our controllers are somewhat less robust, but demonstrate more natural responses during lateral perturbations. This observation is in correspondence with the fact that their framework uses PD control without neural delay for coronal and upper-body balance, while our framework uses muscles with neural delay for all feedback paths.

6.3 Comparisons

We also examine the effect of specific components of our framework by comparing it to versions in which these components are disabled.

Disabling Biomechanical Constraints We examine the effect of the dynamic muscle model and neural delay present in our framework. More specifically, we remove the delay for our feedback signals, and omit activation and contraction dynamics by setting the output muscle force to $F_{\text{M}} = u\text{F}_{\text{max}}$. For effort optimization, we use $E_{\text{effort}}(t) = \sum F_{\text{M}}$.

Together with our Jacobian transpose based control method, the altered controller becomes similar to a joint torque PD controller, with the exception that constant forces still have variable moment arms. The results in the supplementary video material clearly demonstrate increased stiffness and high-gain torque patterns in animations where biomechanical constraints are removed.

Disabling Muscle Routing Optimization We examine the effect of the optimization of muscle routing by fixing each muscle point to the center of its allowed area and removing the routing parameters from the optimization. With this restriction, optimized controllers either fail to produce robust gaits, or demonstrate clear unnatural artifacts in gait.

This effect is most apparent with some of the wider creatures (as can be seen in the supplementary video material), where the effective placement of the optimized lateral gluteus muscles allows the character to smoothly pivot around its stance leg, thus achieving a more natural gait. In another example, which shows a humanoid with wide hips and a thin upper-body, the optimized placement of the lower abdominal muscles aids in achieving upper-body stability.

7 Conclusions

We have introduced a flexible framework for muscle-based locomotion of bipedal creatures. The method has the versatility to support various creatures, a range of speeds, turning behavior, and robustness to external perturbations and unanticipated variations in terrain slope. Key elements include the optimization of muscle routing and the use of muscle-based approximations to Jacobian transpose control. Together, these allow for flexible and robust fully-3D muscle-driven locomotion for a variety of bipedal creatures.

The current method still has limitations. Compared to the results of Wang et al. [2012], our human walking and running motions are of somewhat lesser fidelity, especially for the upper-body. This can be partially explained by the absence of target arm features in

our humanoid models. We have left out such targets in favor of a generic approach, but researchers focusing on a more faithful human gait can easily reintroduce these (and other) domain-specific elements. Another important constituent is the fact that Wang et al. [2012] use PD-driven torques without delay for all coronal and upper-body motion, which gives their controller exceptional lateral balance and upper-body stability. This allows their character to move more solidly in a straight line (especially during running motions), but makes it respond less naturally to perturbations. Apart from these aspects, our lower-body walking motions are very close to their state-of-the-art result. We witness a similar near-passive knee usage during swing, as well as a natural build-up of the ankle plantarflexion moment during stance. This is remarkable given the fact that we left out several of the domain-specific feedback rules introduced by Geyer and Herr [2010].

A fundamental question shared by much of the work in this area is that of what to do when an optimization does not produce the desired results. It can be difficult to know whether to attribute the outcome to implementation errors, the optimization method finding a local minimum, the weighting of objective function terms, the given muscle routing templates, the creature morphology, or limitations of the control architecture. In practice, we have found the modular, parameterized structure of our creatures to be helpful in gaining a deeper understanding of how these various factors help shape the resulting motion patterns. The development of an improved set of authoring tools remains an important direction for future work.

Muscle-based control provides a lower-level model for generating creature motion than previous torque-based control methods, and much lower-level (more detailed) than that of kinematic models of motion. This has the potential to create significantly better models of motion, because the constraints imparted by muscle-based control now become implicit in the resulting motions. However, commensurate with this is the disadvantage of having a larger set of parameters that need to be modeled or identified from data, i.e., muscle geometry, muscle maximum forces, and other such parameters. Our results show that optimization can be used as one method to help set these extra parameters, at least for the constrained set of models and motions we have presented here. Within the scope of our framework, we demonstrate that there is a benefit to the muscle model and the muscle geometry optimization. We note, however, that this indicates that muscle models are sufficient, although they still may not always be necessary if high quality results can be achieved using other means (i.e., simpler kinematic or dynamic modeling methods). While Wang et al. [2012] test and document the importance of using muscle models as compared to torque-based models, a more exact characterization of the benefits and limitations of each of these classes of models remains an important subject for future work.

Details of our simulation which could be further improved include: greater fidelity for the modeling joints such as the knees, ankles, and shoulders; more accurate muscle path wrapping models that interact with the skeleton geometry; giving further thought to the detail with which the target feature trajectories need to be modeled; the addition of anticipatory feed-forward control to the architecture; and the use of alternate dynamics simulators such as *OpenSim*, which have been thoroughly tested in the context of other biomechanics research efforts. It would be interesting to investigate the extent to which the muscle geometry optimization can be used to predict the insertion and attachment points of human musculature. An analysis of the motion with respect to the actions of antagonistic muscle pairs would also be helpful in terms of understanding the solution space. Lastly, there is a need to investigate a wider repertoire of motions, including speed transitions and more aggressive balance recovery behaviors.

Acknowledgments

We wish to thank Frans C. T. van der Helm and the reviewers for their valuable feedback. This research was supported by the GALA project, funded by the European Union in FP7. Michiel van de Panne was supported by NSERC and GRAND.

References

ACKERMANN, M., AND VAN DEN BOGERT, A. J. 2012. Predictive simulation of gait at low gravity reveals skipping as the preferred locomotion strategy. *Journal of Biomechanics 45*, 7, 1293–8.

ANDERSON, F., AND PANDY, M. 2001. Dynamic optimization of human walking. *Journal of Biomechanical Eng. 123*, 381.

COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2009. Robust Task-based Control Policies for Physics-based Characters. *ACM Trans. on Graphics 28*, 5.

COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2010. Generalized biped walking control. *ACM Trans. on Graphics 29*, 4.

COROS, S., KARPATHY, A., JONES, B., REVERET, L., AND VAN DE PANNE, M. 2011. Locomotion skills for simulated quadrupeds. *ACM Trans. on Graphics 30*, 4.

DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Interactive simulation of stylized human locomotion. *ACM Transactions on Graphics (SIGGRAPH) 27*, 3 (Aug.), 1–10.

DA SILVA, M., ABE, Y., AND POPOVIC, J. 2008. Simulation of human motion data using short-horizon model-predictive control. *Computer Graphics Forum 27*, 2, 371–380.

DE LASA, M., MORDATCH, I., AND HERTZMANN, A. 2010. Feature-Based Locomotion Controllers. *ACM Trans. on Graphics 29*, 3.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *ACM SIGGRAPH Papers*, 251–260.

GEIJTENBEEK, T., AND PRONOST, N. 2012. Interactive Character Animation Using Simulated Physics: A State-of-the-Art Review. *Computer Graphics Forum 31*, 8 (Dec.), 2492–2515.

GEIJTENBEEK, T., VAN DEN BOGERT, A. J., VAN BASTEN, B. J. H., AND EGGES, A. 2010. Evaluating the physical realism of character animations using musculoskeletal models. In *Motion in Games*. Springer, 11–22.

GEIJTENBEEK, T., PRONOST, N., AND VAN DER STAPPEN, A. 2012. Simple Data-Driven Control for Simulated Bipeds. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, The Eurographics Association, Lausanne, Switzerland, P. Kry and J. Lee, Eds., 211–219.

GEYER, H., AND HERR, H. 2010. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE transactions on neural systems and rehabilitation engineering 18*, 3 (June), 263–73.

GEYER, H., SEYFARTH, A., AND BLICKHAN, R. 2003. Positive force feedback in bouncing gaits? *Proc. of the Royal Society of London. Series B: Biological Sciences 270*, 1529, 2173–2183.

GEYER, H., SEYFARTH, A., AND BLICKHAN, R. 2006. Compliant leg behaviour explains basic dynamics of walking and running. *Proc. of Biological sciences / The Royal Society 273*, 1603 (Nov.), 2861–7.

GRZESZCZUK, R., AND TERZOPOULOS, D. 1995. Automated learning of muscle-actuated locomotion through control abstraction. In *ACM SIGGRAPH Papers*, 63–70.

HANSEN, N. 2006. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation*, 75–102.

HECKER, C., RAABE, B., ENSLOW, R. W., DEWEESE, J., MAYNARD, J., AND VAN PROOIJEN, K. 2008. Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. on Graphics 27*, 3 (Aug.), 1.

HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *ACM SIGGRAPH Papers*, 71–78.

IJSPEERT, A. J., CRESPI, A., RYCZKO, D., AND CABELGUEN, J.-M. 2007. From swimming to walking with a salamander robot driven by a spinal cord model. *Science (New York, N.Y.) 315*, 5817 (Mar.), 1416–20.

JAIN, S., AND LIU, C. 2011. Modal-space control for articulated characters. *ACM Trans. on Graphics 30*, 5.

JAIN, S., YE, Y., AND LIU, C. K. 2009. Optimization-based interactive motion synthesis. *ACM Trans. on Graphics 28*, 1.

KRY, P., REVERET, L., FAURE, F., AND CANI, M.-P. 2009. Modal Locomotion: Animating Virtual Characters with Natural Vibrations. *Computer Graphics Forum 28*, 2 (Apr.), 289–298.

KWON, T., AND HODGINS, J. 2010. Control systems for human running using an inverted pendulum model and a reference motion capture sequence. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 129–138.

LASZLO, J., VAN DE PANNE, M., AND FIUME, E. 1996. Limit cycle control and its application to the animation of balancing and walking. In *ACM SIGGRAPH Papers*, 155–162.

LEE, Y., KIM, S., AND LEE, J. 2010. Data-driven biped control. *ACM Trans. on Graphics 29*, 4 (July), 129.

LIU, C. K., HERTZMANN, A., AND POPOVIĆ, Z. 2005. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics 24*, 3, 1071.

LIU, L., YIN, K., VAN DE PANNE, M., AND GUO, B. 2012. Terrain Runner: Control, Parameterization, Composition, and Planning for Highly Dynamic Motions. *ACM Trans. on Graphics 31*, 6 (Nov.), 1.

MAUFROY, C., KIMURA, H., AND TAKASE, K. 2008. Towards a general neural controller for quadrupedal locomotion. *Neural networks : the official journal of the International Neural Network Society 21*, 4 (May), 667–81.

MORDATCH, I., DE LASA, M., AND HERTZMANN, A. 2010. Robust Physics-Based Locomotion Using Low-Dimensional Planning. *ACM Trans. on Graphics 29*, 4.

MUICO, U., LEE, Y., POPOVIĆ, J., AND POPOVIĆ, Z. 2009. Contact-aware nonlinear control of dynamic characters. *ACM Trans. on Graphics 28*, 3 (July).

MUICO, U., POPOVIĆ, J., AND POPOVIĆ, Z. 2011. Composite control of physically simulated characters. *ACM Trans. on Graphics 30*, 3 (May).

PANDY, M., ANDERSON, F., AND HULL, D. 1992. A parameter optimization approach for the optimal control of large-scale musculoskeletal systems. *Journal of Biomechanical Engineering, Transactions of the ASME 114*, 4, 450–460.

RAIBERT, M. H., AND HODGINS, J. K. 1991. Animation of dynamic legged locomotion. *ACM SIGGRAPH Computer Graphics 25*, 4 (July), 349–358.

SIMS, K. 1994. Evolving virtual creatures. In *ACM SIGGRAPH Papers*, 15–22.

SMITH, R., 2006. Open Dynamics Engine User Guide v0.5.

SOK, K., KIM, M., AND LEE, J. 2007. Simulating biped behaviors from human motion data. *ACM Trans. on Graphics 26*, 3, 107.

SUEDA, S., KAUFMAN, A., AND PAI, D. K. 2008. Musculotendon simulation for hand animation. *ACM Trans. on Graphics 27*, 3, 83.

SUNADA, C., ARGAEZ, D., DUBOWSKY, S., AND MAVROIDIS, C. 1994. A coordinated Jacobian transpose control for mobile multi-limbed robotic systems. In *IEEE Int. Conf. on Robotics and Automation*, 1910–1915.

TAGA, G. 1995. A model of the neuro-musculo-skeletal system for human locomotion. *Biological Cybernetics 73*, 2, 97–111.

TAN, J., GU, Y., TURK, G., AND LIU, C. 2011. Articulated swimming creatures. *ACM Trans. on Graphics 30*, 4, 58.

THELEN, D., ANDERSON, F., AND DELP, S. 2003. Generating dynamic simulations of movement using computed muscle control. *Journal of Biomechanics 36*, 321–328.

TSAI, Y.-Y., LIN, W.-C., CHENG, K. B., LEE, J., AND LEE, T.-Y. 2010. Real-time physics-based 3d biped character animation using an inverted pendulum model. *Visualization and Computer Graphics, IEEE Transactions on 16*, 2, 325–337.

TSANG, W., SINGH, K., AND FIUME, E. 2005. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, ACM, 319–328.

WAMPLER, K., AND POPOVIĆ, Z. 2009. Optimal gait and form for animal locomotion. *ACM Trans. on Graphics 28*, 3 (July), 1.

WAMPLER, K., POPOVIĆ, J., AND POPOVIĆ, Z. 2013. Animal Locomotion Controllers From Scratch. *Computer Graphics Forum 32*.

WANG, J., FLEET, D., AND HERTZMANN, A. 2009. Optimizing walking controllers. *ACM Trans. on Graphics 28*, 5, 168.

WANG, J., FLEET, D., AND HERTZMANN, A. 2010. Optimizing Walking Controllers for Uncertain Inputs and Environments. *ACM Trans. on Graphics 29*, 4.

WANG, J., HAMNER, S., DELP, S., AND KOLTUN, V. 2012. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Trans. on Graphics 31*, 4, 25.

WU, J.-C., AND POPOVIC, Z. 2010. Terrain-Adaptive Bipedal Locomotion Control. *ACM Trans. on Graphics 29*, 4.

YE, Y., AND LIU, C. 2010. Optimal feedback control for character animation using an abstract model. *ACM Trans. on Graphics 29*, 4 (July), 74.

YIN, K. K., LOKEN, K., AND VAN DE PANNE, M. 2007. Simbicon: Simple biped locomotion control. *ACM Trans. on Graphics 26*, 3, 105.

ZAJAC, F. E. 1989. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical reviews in biomedical engineering 17*, 4, 359–411.