

# Optimizing Walking Controllers

Jack M. Wang    David J. Fleet    Aaron Hertzmann

University of Toronto

## Abstract

This paper describes a method for optimizing the parameters of a physics-based controller for full-body, 3D walking. A modified version of the SIMBICON controller [Yin et al. 2007] is optimized for characters of varying body shape, walking speed and step length. The objective function includes terms for power minimization, angular momentum minimization, and minimal head motion, among others. Together these terms produce a number of important features of natural walking, including active toe-off, near-passive knee swing, and leg extension during swing. We explain the specific form of our objective criteria, and show the importance of each term to walking style. We demonstrate optimized controllers for walking with different speeds, variation in body shape, and in ground slope.

**Keywords:** Physics-based animation, controller synthesis, human motion, optimization.

## 1 Introduction

Locomotion is an essential skill for simulated characters, but one for which designing controllers is particularly difficult. The control space is high-dimensional, the dynamics are nonlinear and discontinuous due to contact, and infeasible controllers (i.e., that trip and fall) are all too common. While robust walking controllers have recently been designed [Kim et al. 2007; Yin et al. 2007], they produce gaits that appear unnatural. Indeed, it remains extremely challenging for either humans or computers to define robust controllers for natural-looking walking.

This paper introduces a parameter optimization procedure for full-body, 3D walking controller synthesis. Controllers may be optimized for characters of varying body shape, and, optionally, to achieve user-specified walking speeds and/or step lengths. The resulting gaits exhibit key properties of natural walking, including, for example, energy efficiency, a strong *toe-off* as support is transferred from one foot to the other, a nearly passive knee during leg swing, leg extension prior to *heel-strike*, torso lean, and the anti-symmetric phase of arm swing. We advance the state-of-the-art by demonstrating results that both capture these important features and are robust enough to tolerate minor environmental disturbances. Unlike many previous methods, the system does not require any motion capture data nor reference trajectories.

The control parameterization is a version of SIMple BIped CONtrol (SIMBICON) [Yin et al. 2007], with modifications to allow more

<http://www.dgp.toronto.edu/~jmwang/optwalk/>

realistic motion. The objective function is a weighted sum of several terms, many of which are inspired by biomechanical properties of human walking. We demonstrate how each of these terms contributes to creating human-like qualities of motion. The optimization is initialized with a walker that is unstable, but roughly captures observed features of human walking. Because the approach does not rely on motion capture data, there is no restriction that the styles of walking follow any particular mocap database.

## 2 Related Work

Many applications of controllers in graphics have been based on hand-tuned state machine models, including full-body walking (e.g., [Laszlo et al. 1996; Faloutsos et al. 2001]). Perhaps the most important such model is the recent SIMBICON controller by Yin et al. [2007]. While remarkably robust, and capable of producing several styles, it produces motions that differ in several key ways from human walking. For example, it relies significantly on the hip to generate forward momentum, while normal human walking relies heavily on the ankle, with a near-passive knee [Novacheck 1998]. As a consequence, SIMBICON produces a marching-like gait.

The original control optimization methods in animation were applied to low-dimensional virtual characters [van de Panne and Fiume 1993; Sims 1994; van de Panne and Lamouret 1995], where the search space is much smaller than full-body walking. Grzeszczuk et al. [1995; 1998] demonstrate control optimization for high-dimensional marine animals, where smooth dynamics (without ground contacts) lead to smoother objective functions than for ground-based locomotion. Sharon and van de Panne [2005] optimize planar humanoid characters to match the style of reference motion, but their results suggest that this approach has pros and cons similar to Yin et al. [2007]. Inspired by passive dynamic walking, Collins et al. [2005] demonstrate low-energy robots that produce human-like walking gaits. The controller for one such robot is designed by online optimization [Tedrake et al. 2004], but its physical design is stable and its control input space is low-dimensional.

For 3D humanoid models, Hodgins and Pollard [1997] adapt stable controllers from one character to different characters by searching over a small number of high-level control parameters. More recently, optimization-based methods are used to adapt walking controllers for more difficult tasks [Yin et al. 2008], or to create solutions for constrained terrain navigation [Coros et al. 2008]. In both cases, a hand-tuned robust walker is available as initialization, only a relatively small subset of task-relevant parameters are optimized, and the basic style of motion is not modified. In contrast, our focus is on the style of walking with a smaller set of tasks. We begin with a rough, unstable controller and show how to significantly improve walking style and robustness.

Another approach is to employ motion capture data to define the desired controller [Sok et al. 2007; da Silva et al. 2008; Muico et al. 2009; Tsai et al. in press]. This frees the designer from specifying the style of the motion, only requiring the maintenance of balance. The use of mocap data allows for high-quality human-like results to be simulated, as recently demonstrated by Muico et al. [2009] on an impressive array of locomotion skills. However, such methods are limited in their ability to generalize to situations where no mocap data are available.

Trajectory optimization (of pose, torque, or muscle excitation) has a long history in both graphics and biomechanics. The spacetime constraints method [Witkin and Kass 1988] solves for a sequence of poses subject to user constraints, while minimizing an approximation to energy consumption. By learning an objective function from data, Liu et al. [2005] extend the method to generate full-body walking motion given footstep constraints. More recently, Wampler and Popović [2009] combine traditional spacetime constraints with Covariance Matrix Adaptation (CMA) [Hansen 2006], a derivative-free optimization algorithm also used in this paper, to generate plausible locomotion gaits for a variety of virtual creatures. In biomechanics, Anderson and Pandy [2001] recover muscle excitation trajectories during a walk cycle by minimizing metabolic energy over distance travelled. A main difference between trajectory optimization and our work is that we aim to recover controllers that work beyond the duration for which they were optimized, possibly subject to minor environmental disturbances. In contrast, the recovered forces from trajectory optimization are not generally robust to any change in environmental interactions if used to control new simulations.

### 3 Character Model and Controllers

Our character model and control parameterization are based on SIMBICON, with a few small but important differences. The character model has 30 internal degrees-of-freedom (DOFs), 28 of which are identical to Yin et al. [2007]. We add toe blocks, which are connected to the feet by hinge joints. Toes provide more flexibility during landing and ankle push-off (also called toe-off). The link locations and mass distributions are based on Wooten and Hodgins [1996]. We scale the links and masses according to the dimensions of the character skeleton, and apply reasonable joint limits.

**Single-state controller.** As in SIMBICON, the walking controller is a finite-state machine. Each state contains proportional-derivative (PD) controllers for each joint and a balance feedback controller. The controller for each joint DOF includes gain and damping coefficients ( $k_p, k_d$ ), as well as a target angle ( $\theta_d$ ). At each simulation time-step, a torque ( $\tau$ ) for each joint DOF is generated according to

$$\tau = k_p(\theta_d - \theta) + k_d\dot{\theta}, \quad (1)$$

where  $\theta$  and  $\dot{\theta}$  are the current joint angle and angular velocity.

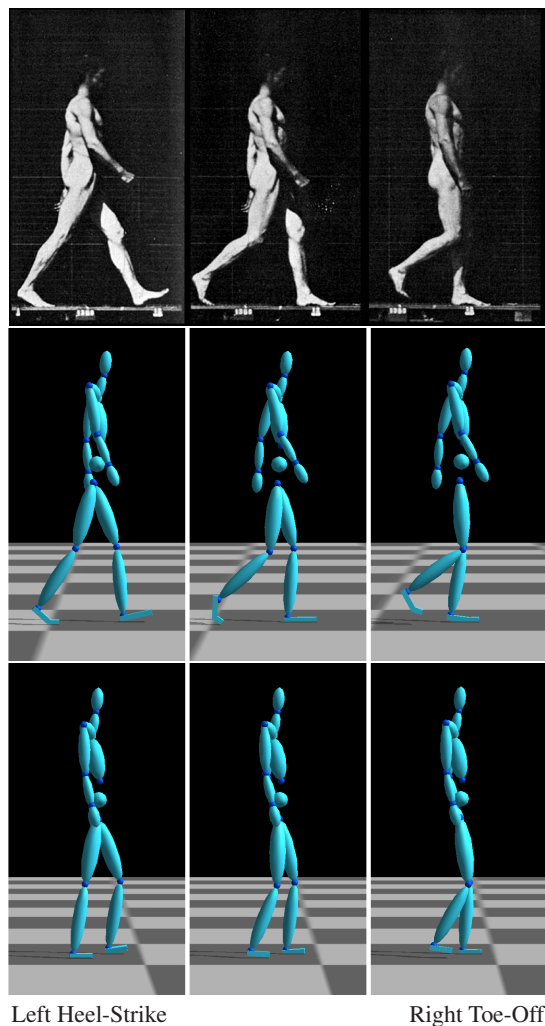
The input angles ( $\theta$ ) to the PD controllers are expressed in local coordinate systems, except for the hip and ankle. The hip angles operate in the world frame and are mapped to control the torso orientation, adjusted by balance feedback parameters ( $c_d, c_v$ ) [Yin et al. 2007]<sup>1</sup>. The world frame orientation of a body part is defined by measuring angles of their up and forward vectors projected in the coronal, sagittal, and transverse planes. Unlike SIMBICON, our controllers servo the ankles in the world frame, since world orientation is crucial for ensuring ground clearance after toe-off, as well as for landing at a good angle to allow weight transfer.

We couple the target angle of the arm DOF in the sagittal plane to the hip angles in the sagittal plane, so that the target angle is

$$\theta_{larm} = \alpha_{arm}(\theta_{rhip} - \theta_{lhip}), \quad \theta_{rarm} = -\theta_{larm}, \quad (2)$$

where  $\alpha_{arm}$  is a scale factor. This allows the model to synchronize arm swing with the legs, but good arm swing still depends on a good selection of  $\alpha_{arm}$  and arm spring-damper constants.

<sup>1</sup>Since the stance  $hip_z$  DOF serves to rotate the body towards the desired facing angle, we only enable it when the stance foot is firmly planted (more than 3 points in contact on the foot or toe), otherwise it is servoed to zero.



**Figure 1:** Comparison of our model to human and SIMBICON from the left heel-strike to right toe-off stages of walking. **Top:** Man walking [Muybridge 1887]. **Middle:** One of our optimized controllers. **Bottom:** Our implementation of SIMBICON. Note the more conservative style in the bottom row, which keeps the right foot relatively flat, whereas our result and the human show significant rotation. Also note that the left leg in the left column does not reach full extension in the bottom row.

**State machine and transitions.** The finite state machine contains four states, corresponding to basic phases of walking (Figure 2). State 0 begins at foot strike, and continues as the swing leg lifts off and swings forward. The transition to state 1 occurs when the signed horizontal distance (in the sagittal plane) between the center-of-mass (COM) of the body and the ankle of the stance foot exceeds a threshold  $c_{trans}$ . This is motivated by our observations of when stance ankle push-off appears to occur. Notably, this differs from SIMBICON which uses a time-based transition.

During state 1, the swing leg prepares for landing, and the stance ankle push-off begins. The transition to state 2 occurs when the swing foot makes contact with the ground, provided that the swing ankle global orientation (in the sagittal plane) exceeds 0.1 radians. Without this condition, the controller must lift the swing leg to an artificial height to ensure ground clearance. State 0 may also transition directly to state 2 if contact occurs (although this usually indicates a poor controller). States 2 and 3 are left/right reflected versions of states 0 and 1, with mirrored parameters.

State #	0	1	2	3
Left Foot	Swing	Swing	Stance	Heel-off
Right Foot	Stance	Heel-off	Swing	Swing

Contact

**Figure 2:** The state machine for the walking controller consists of four states, corresponding to ground contact and swing phases. The transition from state 0 to state 1 occurs approximately when the left foot passes the right foot, and the transition to state 2 occurs when the left foot makes contact with the ground. The states are left/right symmetric, with states 2 and 3 mirroring states 0 and 1.

**Start state.** The start state of the physical simulation is specified by 6 global DOFs, 30 joint DOFs, and their generalized velocities (72 DOFs in total). The start state is optimized along with the controller, and is manually initialized (see Table 2) to when the left leg is in the middle of its swing phase, prior to the transition from state 0 to state 1.

In principle, the start state and the controller parameters total over 200 DOFs in the system. However, since we focus on straight walking in the positive  $x$  direction in this paper, we fix DOFs and other parameters that are unlikely to contribute to the task. More specifically, joint DOFs (including PD control target angles) that rotate with respect to the local  $x$  and  $z$  axis are fixed to zero, to discourage unnecessary motion in the coronal and transverse planes. The back joint is an exception, where rotation with respect to  $z$ -axis is free to be optimized, which is necessary for trunk rotation in the gait. The target angles for the toe joints are fixed to zero except for the stance foot during states 1 and 3. For the global DOFs in the start state,  $x$  and  $y$  positions can naturally be set to any value. Since the controller would not be walking straight if it deviates away from the  $y$ - $z$  plane, we fix the start state global rotation (and angular velocity) with respect to  $x$  and  $z$  axis, as well as velocity in the  $y$ -direction to zero. With these constraints, the start state and the control parameters comprise a 184 dimensional search space.

## 4 Controller Optimization

Optimizing a controller involves searching for control parameters and a start state that together produce good character simulations. The quality of the generated motion is measured with an objective function that evaluates simulations of duration 10 seconds ( $T$  simulation time-steps). The objective function comprises a weighted combination of terms motivated by task constraints and biomechanical features of human walking.

### 4.1 Objective Function

An obvious way to define an objective function is with a weighted sum of quadratic penalty terms on quantities such as total power consumption and deviation from a target speed. In practice, finding suitable weights for such terms is extremely difficult. For example, if chosen poorly, then even when the target speed is nearly achieved, optimization might continue to ignore the energy term in favor of imperceptible speed refinements. Instead, we employ a weighted combination of objectives that do not penalize small differences from targets. Specifically, we define a thresholded quadratic as

$$Q(d; \epsilon) = \begin{cases} d^2, & \text{if } |d| > \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

This objective assigns a penalty of zero when the distance  $d$  is below a threshold  $\epsilon$ , but applies steep penalties beyond this threshold. This formulation is akin to specifying hard constraints on the motion to optimize power and stability related terms, subject to constraints that speed and other quantities fall within  $\epsilon$  of target values. Unlike hard constraints, however, including the soft penalty component allows us to avoid the difficult problem of finding a feasible initialization (and ensuring that it exists).

**User gait constraints.** A user may specify high-level requirements on the average forward speed ( $v_x$ ) or step length ( $s$ ) of the simulation. This is done by penalizing the differences between these values and user-specified targets ( $\hat{v}_x, \hat{s}$ , respectively):

$$E_{user} = Q(v_x - \hat{v}_x; \epsilon_{vel}) + Q(s - \hat{s}; \epsilon_{step}), \quad (4)$$

Both of these terms are optional.

**Required gait constraints.** There are several properties of gait that we find essential to the style and stability of the simulated walking motions. First, because we optimize for walking in the positive  $x$  direction, significant deviations in the  $y$  and  $z$  directions of motion are undesirable. Accordingly, they are penalized with the following objective:

$$E_{vel} = Q(v_y; \epsilon_{vel}) + \lambda_{vel} [Q(v_{0x} - v_x; \epsilon_{vel}) + Q(v_z; 2\epsilon_{vel})], \quad (5)$$

where  $v_y$  and  $v_z$  are the average simulation velocities (per second) of the COM in the  $y$  and  $z$  directions, respectively, and  $v_{0x}$  is the velocity of the start state in the  $+x$  direction. This constraint encourages  $v_y$  and  $v_z$  to be small, and the start velocity  $v_{0x}$  to be similar to the average  $x$  velocity of the simulated motion.<sup>2</sup>

Since we only optimize the controller for simulations of a fixed duration (10 seconds), we need to encourage solutions that are more likely to lead to stable walk cycles when simulated beyond that duration. Two major sources of instability are swing foot toe-stubbing and toe-off before the stance foot is firmly planted on the ground. We discourage these situations with the following objective:

$$E_{land} = \frac{1}{T} \sum_{t=1}^T (stance_t + stubbed_t), \quad (6)$$

where the summation is over all simulation steps  $t$ . If, during state 1 or 3 (toe-off), neither the stance toe nor the stance foot have 3 or more points of contact with the ground, then  $stance_t = 1$ ; it is set to zero otherwise. We set  $stubbed_t = 1$  at any time when the top of the swing toe is in contact with the ground.

Similarly, we define

$$E_{fail} = \frac{1}{T} \sum_{t=1}^T failed_t, \quad (7)$$

where for all states, if the COM falls below 0.7 m (i.e., the simulated character has fallen down) then  $failed_t = 1$ . With the exception of character “long arm” (see Figure 8(bottom)), which fails if the COM falls below 0.4 m.

Our fourth gait objective encourages left-right symmetric timing of the controller, i.e.,

$$E_{sym} = Q(\Delta t_0 - \Delta t_2; \epsilon_t) + Q(\Delta t_1 - \Delta t_3; \epsilon_t), \quad (8)$$

where  $\Delta t_i$  is the average duration of state  $i$  during the walk. This term,  $E_{sym}$ , requires left and right strides to have approximately the same duration. This helps to avoid spurious local minima producing asymmetric gaits.

<sup>2</sup>We replace  $v_x$  with  $\hat{v}_x$  if the latter is available.

**Head and body constraints.** The angular momentum of the body about its COM is typically very small in human walking [Herr and Popović 2008], and has been shown to assist with balancing [Kudoh et al. 2006; Macchietto et al. 2009]. In particular, a mechanical effect of arm swing is to reduce torso torques about the vertical axis induced by the lower body [Li et al. 2001]. Motivated by this, we include the following objective:

$$avgL = \frac{1}{T} \sum_{t=1}^T \dot{L}_t^2, \quad (9)$$

$$E_{ang} = Q(\sqrt{avgL}; \sqrt{\epsilon_{ang}}), \quad (10)$$

where  $\dot{L}_t$  is the derivative of the normalized angular momentum [Herr and Popović 2008] about the COM in the vertical direction at time step  $t$ , computed using finite differences. We find that this term helps to prevent unnatural arm swing, where the arms and legs are badly out of phase.

In natural human walking, the lateral and vertical motions of the head are typically smooth with small amplitudes. This helps to stabilize the visual and vestibular systems [Pozzo et al. 1990]. We include an objective that encourages the head to exhibit a fixed orientation and a constant forward velocity. Let  $\Theta_i = [\theta_{cor}, \theta_{sag}, \theta_{trans}]$  represent the head world frame orientation at simulation step  $i$ . The objective to stabilize the head motion is then

$$E_{head} = Q(\sqrt{\sigma_{head}}; \sqrt{\epsilon_{head}}) + \frac{\lambda_{head}}{T} \sum_{t=1}^T orient_t, \quad (11)$$

where  $\sigma_{head}$  is the standard deviation of the head velocity in the  $x$  direction during the walk,  $orient_t$  is a binary variable which is set to 1 when  $\|\Theta_t\|_2 > 0.1$ .

**Efficiency and power terms.** One notable biomechanical property of human walking is its efficiency [Alexander 2003; Collins et al. 2005]. Indeed, the nearly-passive nature of leg and arm swing are characteristic of a natural walking gait. To encourage an efficient gait, we penalize the sum of the squared torques over the duration of the simulation:

$$E_{power} = \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^m \tau_{tj}^2, \quad (12)$$

where  $\tau_{tj}$  is the torque output at time step  $t$  for joint DOF  $j$ . If the target velocity is unspecified, we replace  $E_{power}$  with  $\frac{E_{power}}{v_x}$ , to approximate the cost of transport [Collins et al. 2005].

It is also well-known that, unlike human running, human walking is powered more by the ankle than the hip [Novacheck 1998]. To encourage a natural distribution of torques in the hip, knee and ankle, we encourage the ratio of power outputs from these joints to match those observed in humans. Specifically, let  $\vec{r}$  be the relative magnitudes of the power output from the hip, knee, and ankle, produced by the controller; i.e.,

$$\vec{r} = \frac{[P_{hips}, P_{knees}, P_{ankles}]}{P_{hips} + P_{knees} + P_{ankles}}, \quad (13)$$

where

$$P_{DOF} = \frac{1}{T} \sum_{t=1}^T \sum_{j \in DOF} \tau_{tj}^2. \quad (14)$$

We penalize the deviation of  $\vec{r}$  from the empirical power ratio (hip to knee to ankle),  $\vec{r}_{walk} = [0.43, 0.04, 0.53]$ , observed in humans [Novacheck 1998].

$$E_{ratio} = \|\vec{r} - \vec{r}_{walk}\|_2 \quad (15)$$

Note that this term encourages a nearly passive knee.

**Complete objective.** The complete objective function for walking is given by

$$E = \sum_s w_s E_s, \quad (16)$$

where  $s \in \{user, vel, land, fail, sym, ang, head, power, ratio\}$ . We use the following parameters for all experiments:  $w_{user} = 100$ ,  $w_{vel} = 100$ ,  $w_{land} = 1.2$ ,  $w_{fail} = 120000$ ,  $w_{sym} = 100$ ,  $w_{ang} = 10$ ,  $w_{head} = 100$ ,  $w_{power} = 10^{-5}(70/mass)$ ,  $w_{ratio} = 1$ ,  $\lambda_{vel} = 0.01$ ,  $\lambda_{head} = 0.012$ ,  $\epsilon_{vel} = 0.025$  m/s,  $\epsilon_{step} = 0.025$  m,  $\epsilon_t = 0.025$  s,  $\epsilon_{ang} = 0.05$   $ks^2$ ,  $\epsilon_{head} = 0.1$  m/s.

## 4.2 Simulation

Simulation is performed using the Open Dynamics Engine (ODE, version 0.9), with a simulation frequency of 2400Hz. Ground contact is modeled using the default collision detector with a maximum of 4 points on the toe and 4 points on the foot. Both the toe and foot are modeled as boxes, and the ground is modeled as a plane. The contact parameters are adjusted to simulate a spring-damper system with  $k_p = 75000$ ,  $k_d = 2000$ . We set the coefficient of friction to  $\mu = 10$ , which is higher than physically realistic, but necessary to prevent lateral slipping during toe-off.

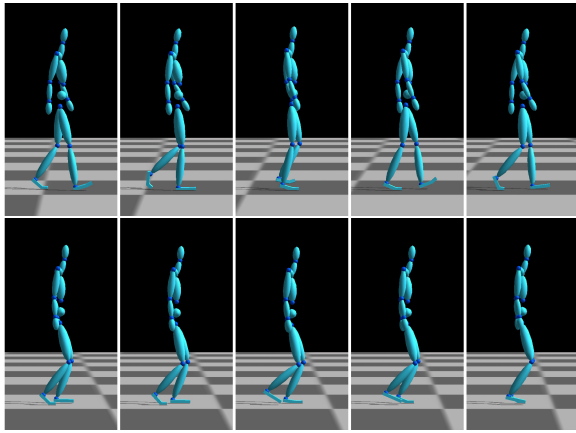
## 4.3 Optimization Algorithm

The optimization problem is high-dimensional, discontinuous, and subject to many local minima. Moreover, each function evaluation involves a simulation in ODE, which runs in approximately real-time. It is important for the method to use as few function evaluations as possible, without the need to evaluate gradients. We tested several different optimization algorithms, and found Covariance Matrix Adaptation (CMA) [Hansen 2006] to work best.

CMA is an iterative algorithm that maintains a Gaussian distribution over parameter vectors. The Gaussian is initialized with a mean (discussed below), and a spherical covariance matrix with diagonal elements equal to  $\sigma^2$ , where  $\sigma$  is a problem-dependent step size parameter.  $M$  random samples are drawn from the Gaussian, and the objective function is evaluated for each. A new Gaussian is constructed using the best  $N$  samples, based on a function of these samples and the old mean. The process is repeated, as the Gaussian converges to a low-objective region. It is assumed that the parameters are scaled to lie roughly between 0 and 1.

We chose  $\sigma = 0.0025$  experimentally. For the number of samples, we use default parameters that are automatically chosen by CMA based on the dimension of our optimization problem. In particular,  $M = 19$ ,  $N = 9$ . The 19 simulations are computed in parallel using a cluster of 19 Intel Xeon 3.8GHz CPUs. In total, 1000 CMA iterations take approximately 3 hours. Heuristically, we stop the optimization either when no better values are discovered for 300 iterations, or when the total number of iterations exceed 3000.

We initialize the CMA optimization with a hand-tuned controller that generates significant ankle push-off in the stance leg during states 1 and 3. The parameters for the initial simulation state (start state) and the PD controllers are included in Table 2. Additionally, the coronal balance feedback parameters are initialized to  $c_d = 0.2$ ,  $c_v = 0.2$ , the sagittal ones are initialized to zero. The arm swing and state transition parameters are initialized to  $\alpha_{arm} = 1.0$ ,  $c_{trans} = 0.01$ . Note that this controller is only tuned to take a few steps in one of the character skeletons (“stocky” in Figure. 8(bottom)). It is not stable and does not walk in a straight line, but is sufficient for initialization. The initial values of  $k_p$  and  $k_d$  are scaled for very heavy and light characters, in proportion to the character’s weight ratio relative to the default character.



**Figure 3:** *Top:* Optimization of “short” (Figure 8(bottom)) walking in 1.0 m/s. *Bottom:* Optimization without  $E_{ratio}$ . The lack of the power ratio term leads to a semi-crouching style.

## 5 Experiments

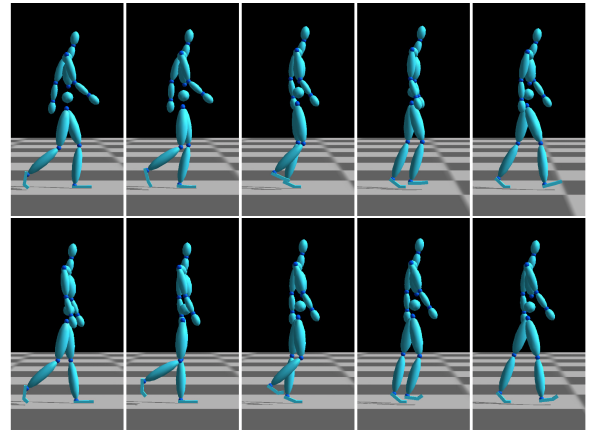
We now describe experiments that demonstrate the approach and test the effects of its various elements; results can be seen more clearly in the accompanying video. Characters used are depicted in Figure 8(bottom).

**Features of human walking.** Our model captures a number of features of normal human walking. Figure 1 shows images of a walking human, frames from a walker generated with our system, and frames from our implementation of SIMBICON [Yin et al. 2007], which represents the state-of-the-art in full-body controllers without mocap. The SIMBICON model employs a hip push-off strategy, whereas our model uses an ankle push-off more similar to human walking. The body synchronization approximates that of the human as well: stance foot heel-off occurs just before swing foot heel-strike. Furthermore, our model captures the passive knee and leg extension of the swing leg, as well as the foot rotation.

**Effects of individual terms.** Next we compare optimizing our walker with and without various energy terms. For example, we optimize the “short” character (Figure 8(bottom)) to walk in 1.0 m/s without a specified step length. Figure 3 compares the model optimized with and without the power ratio term ( $E_{ratio}$ ). Without this term, the power ratio between hip, knee, and ankle in the resulting motion is  $[0.38, 0.21, 0.41]$ ; the knee takes a much heavier load than in human walking. This effect can be observed in the animation as a semi-crouching style which appears unnatural and tiring. In contrast, our model discovers a more relaxed gait, with a power ratio of  $[0.43, 0.05, 0.52]$ , close to our optimization target.

Figure 4 shows our model optimized with and without the angular momentum objective ( $E_{ang}$ ). In this example, we optimized the “stocky” character (Figure 8(bottom)) to walk at 1.8 m/s with a step length of 0.7 m. The resulting motion correctly exhibits in-phase arm-swing, with the arms counter-oscillating with respect to the legs. Without the angular momentum term, the arms are in-phase with the wrong legs, and the walker is unstable. Indeed, the controller does not walk successfully much beyond the optimized duration of 10 seconds, whereas the controller learned with  $E_{ang}$  walks for at least 100 seconds.

We find that optimizing without the head stabilization term ( $E_{head}$ ) leads to jerky upper-body motion (see video). We test optimizing without the  $E_{landing}$  term, which prefers the stance foot to be stably

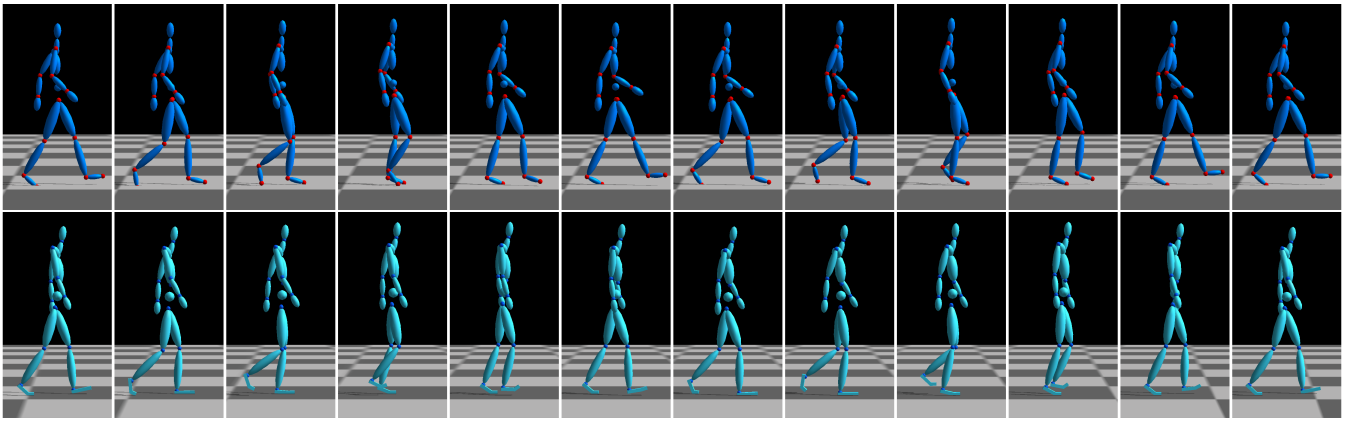


**Figure 4:** *Top:* Optimization of “stocky” (Figure 8(bottom)) walking in 1.8 m/s, step length 0.7 m. *Bottom:* Optimization without  $E_{ang}$ . Note the difference in arm swing.

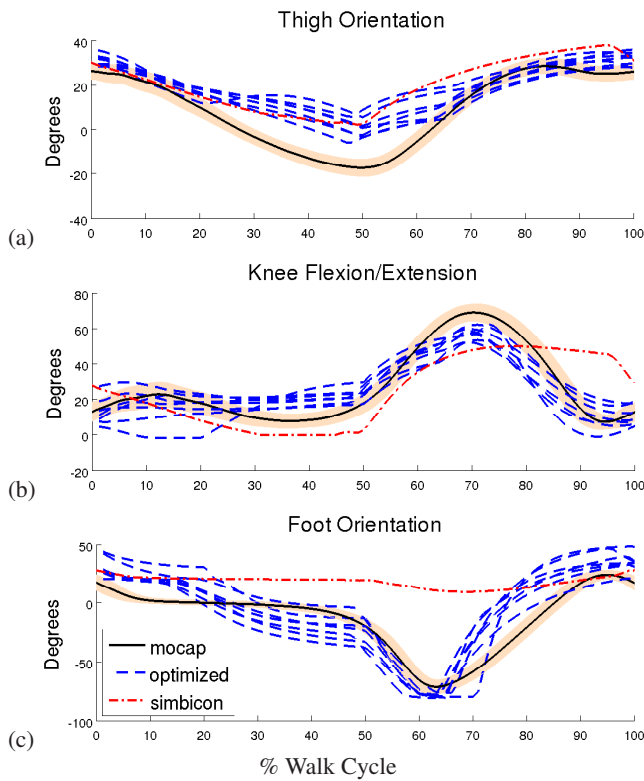
planted before push-off. Without this term, foot landing and roll is jerky and abrupt, whereas including the  $E_{landing}$  term makes the foot land and roll forward smoothly. The  $E_{power}$  term serves to constrain DOFs that are not directly influenced by the other terms to behave in a relaxed fashion. This is particularly relevant to the upper body; as shown in the video, omitting this term leads to jerky arm motion.

**Comparison with motion capture data.** We further evaluate our results by comparing our optimized controller output with motion capture data. The data comprise a walk cycle from each of 115 subjects, all asked to walk at a comfortable speed. Figure 6 compares the thigh orientation, knee flexion/extension, and foot orientation for 7 of our optimized controllers and the mean curve from the mocap dataset. We see that the minimum hip orientation (Figure 6(a)) in the mocap data is smaller than both of the physically simulated gaits. A similar, though less pronounced, effect occurs in the knee flexion/extension plot (Figure 6(b)) as well, where the optimized controllers tend to both flex and extend less than the mocap average. Figure 5 visually compares one of the optimized controllers with a similar mocap walk cycle. The smaller range of hip and knee motions seem to lead to a smaller step length. Two major differences between our optimized controllers and SIMBICON are knee extension at heel-strike (i.e., Figure 6(b) at 0 and 100%), and the foot trajectories plotted in Figure 6(c). In both cases, motions generated by the optimized controllers are much closer to mocap. However, the foot orientation plot shows our optimized controllers tend to push-off with the ankle earlier than the mocap.

**Optimizing SIMBICON.** In order to separate the effects of our objective function from those of our body model, we apply our optimization to the SIMBICON body model and parameterization. The control parameters that generated the bottom row of Figure 1 is used as initialization. We do not specify the target speed. Instead we penalize the approximate cost of transport ( $\frac{E_{power}}{v_x}$ ) in the objective. We find that our optimization process improves the motion in two ways. First, the optimization discovers anti-symmetric arm swing, even though this control parameterization does not explicitly couple arm swing to leg motion as our model does. Second, the character makes some use of ankle push-off. However, compared to our results, the swing leg does not extend as far, and the ankle push-off is still lacking. Despite the low percentage of knee torque output in the lower body, the knee still appears active.



**Figure 5:** *Top:* Motion capture data for a full walk cycle. *Bottom:* Motion generated by one of the optimized controllers (character “tall” (Figure 8(bottom)), no user gait constraints).



**Figure 6:** Sagittal plane angle versus percentage walk cycle plots. (a) Thigh orientation w.r.t. down vector. (b) Angle between thigh and shin. (c) Ankle orientation w.r.t. front vector. The solid curve and orange region represent the mean and standard deviation of 115 mocap walk cycles. Dashed blue and red curves represent 7 optimized walkers and SIMBICON, respectively.

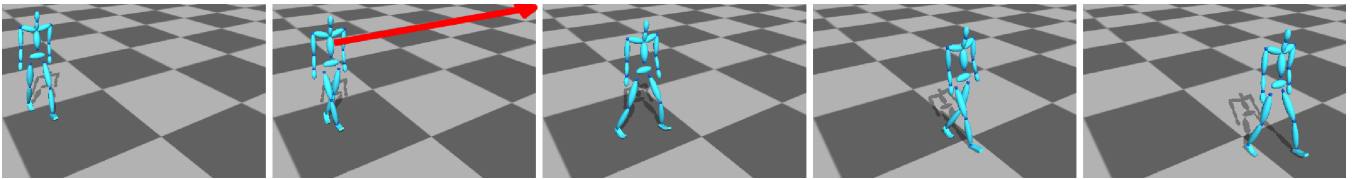
We note that with the same optimization specification (same character, objective) using our model and initialization, we are able to discover a gait with  $E_{power} = 12059$ , much lower than the SIMBICON solution’s  $E_{power} = 33392$ . Moreover, our solution walks with a faster speed of 1.48 m/s versus 1.36 m/s.

**Variation in body shape.** Our optimization can be used to generate controllers for a wide range of body types. We minimize the

approximate cost of transport ( $\frac{E_{power}}{v_x}$ ) to generate walks for all characters in Figure 8(bottom), without specifying target speed and step length. This allows the walker to identify its preferred speed and step length. As shown in the accompanying video, the results retain the qualities of human walking discussed above. A natural relationship between body shape, speed and gait emerges; that is, the tall subject’s preferred gait is much faster (1.79 m/s) than those of the shorter subjects. The overweight subject appears to walk with a more lumbering gait, and the long-armed humanoid keeps its heavy arms still in order to maintain balance.

**Robustness.** We quantify the robustness of some of our controllers via a pushing experiment (see e.g., Figure 7). For each controller, we simulate forward for 40 seconds, and apply a push force ( $F_x, F_y$ ) for 0.4 seconds to the torso COM once every 4 seconds (to allow for recovery time). The controller passes the experiment if it is still walking forward (direction (1, 0)) after 40 seconds. The maximum amount of tolerable push from 8 directions are summarized in Table 1. We observe that the robustness of our controllers varies greatly. The overweight controller is unstable when pushed from the back, but is fairly robust to pushes from the front and sides. All of the optimized controllers tend to be more robust to pushes from the side, than to pushes from the front and back. The experiment also shows that these controllers are less robust than our implementation of SIMBICON, which can stand more than twice as much force than the optimized controllers in some cases. Note that SIMBICON has been reported to withstand force components between 190 N and 340 N [Yin et al. 2007] in a related experiment. We suspect fine tuning our implementation of it can lead to results approaching those numbers as well.

**Changing terrain.** In the spirit of Yin et al. [2008], we obtained uphill walkers for slopes of up to 12 degrees (see Figure 8(top)) by optimizing for a sequence of walkers. This began with optimization for a 3-degree slope, initialized with a walker on flat ground, although it was not run to convergence. For both the uphill walker and initialization, the target step length was set to 0.65 m. We then repeated this process for a sequence of walkers, increasing the slope until 12 degrees. As shown in the video, the controller successfully walks up the slopes, leaning into the slope in a natural way, whereas the level-ground walker fails at the 3-degree slope. Similarly, we optimized a walker for a plane with a lower coefficient of friction. Beginning with an initial walker, we optimized for  $\mu = 0.8$ , and then optimized for  $\mu = 0.3$ . The resulting controller walks more gingerly, at a slower rate.



**Figure 7:** Controller tall (Table 1) reacting to a 200 N, 0.4 s push to the torso in the (0, 1) direction. Our optimized controllers tend to be more robust to pushes from the side, than to pushes from front and back.

direction	short	overweight	tall	short2	stocky	simbicon
(1,0)	75	1	20	10	50	225
(1,1)	75	1	20	10	65	150
(0,1)	100	25	200	50	125	250
(-1,1)	50	50	70	40	75	100
(-1,0)	60	75	120	75	50	120
(-1,-1)	40	40	70	40	50	100
(0,-1)	100	20	200	75	125	275
(1,-1)	75	1	20	10	50	125

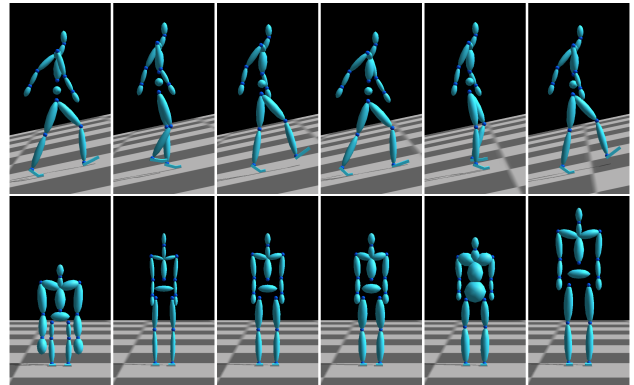
**Table 1:** Maximum disturbance force components (in Newtons) tolerance for some of our controllers and our implementation of SIMBICON, where (1,0) is the forward direction. Controller overweight is optimized for a step length of 0.8 m, short2 for a step length of 0.65 m, and stocky for a step length of 0.7 m and a speed of 1.8 m/s. The others are optimized without user gait constraints.

**Other variations.** As shown in the accompanying video, we can create controllers to walk at specified speed and/or step lengths via  $E_{user}$ , including specifying atypical speed and step length combinations. Alternatively, either speed or step length or both can be determined automatically. Our initial walker in Figure 8(top) is generated by specifying a target step length of 0.65 m without a target velocity. The accompanying video shows more examples of controllers synthesized in this fashion as well as results obtained by altering other terms in the objective function. If we redefine the cost of transport term as  $1/v_x$ , thereby omitting the penalty for total torque, then the objective function prefers motions that walk as fast as mechanically possible, without regard to energy consumption, yielding extremely fast walking.

## 6 Discussion

We have demonstrated a method for optimizing full-body 3D walking control that captures important features of natural walking without relying on mocap data. While the basic idea of optimizing controllers is an old one, our results show that achieving good walking control requires careful choice of body parameterization, controller parameterization, and objective function. In particular, the results illustrate the importance of angular momentum minimization, relative magnitudes of lower-body joint torques, and head stabilization, among others. We believe these observations will be useful for designing more sophisticated controllers and objective functions, especially since our objective function and optimizer are independent of the choice of control parameterization. Our work also illustrates how subtle details in control parameterization — especially at the foot — can make a significant impact on the style of motion.

There are a number of limitations of our method that provide opportunities for future work. Our method requires an expensive optimization procedure, and depends on a reasonable initialization. As our goal is natural-looking human locomotion, the controllers we produce are not as stable as SIMBICON, which could be addressed with a better control parameterization or explicit balance-recovery controllers. The difference is most likely due to the more conserva-



**Figure 8:** Top: The short2 controller (Table 1) adapted to walk up a 12 degree slope. Bottom: Characters used in the paper. From left to right: “long arm” (127 cm, 65 kg), “thin” (165 cm, 41 kg), “short” (165 cm, 57 kg), “stocky” (165 cm, 70 kg), “overweight” (161 cm, 86 kg), “tall” (196 cm, 79 kg).

tive, less human-like strategy taken by SIMBICON that keeps the feet orientated parallel to the ground during the entire gait (Figure 6(c)), making toe-stubbing less likely. Another difference is that, while the balance parameters are tuned for robustness in Yin et al. [2007], our objective function does not explicitly encourage robustness beyond the requirement of walking for 10 seconds.

As is evident from comparison with motion capture data, the motion generated by our controller still differs from human motion in noticeable ways. Figure 6 indicates that the thigh and knee rotation do not straighten as much as the data, especially when the stance leg is behind the COM. Note that when we only specified desired speed without desired step length, the resulting walker typically takes shorter steps than mocap data. This might be explained by the lack of hind leg stretching in our controllers.

Finally, our method does require some parameter tuning in order to achieve reasonable gaits and to achieve the desired style. We believe that adjusting energy parameters will be more intuitive than manually adjusting control parameters. Furthermore, we anticipate that it may be possible to learn the parameters from mocap data.

## Acknowledgements

The authors are indebted to Peter O’Donovan for his tireless video production efforts, and Martin de Lasa and Igor Mordatch for inspiring technical discussions. We thank Prof. Nikolaus Troje for providing the mocap walking data, Prof. Michiel van de Panne and KangKang Yin for providing detailed information regarding SIMBICON, and Nikolaus Hansen for his publicly available CMA implementation. This research is supported in part by CFI, CIFAR, Microsoft Research, NSERC, and Ontario MRI.

Start State			Controller				
DOF	$q_0$	$\dot{q}_0$	State	0		1	
			DOF	$k_p$	$\theta_d$	$k_p$	$\theta_d$
$globpos_x$	free	1.3	$back_{xyz}$	300	0	300	0
$lhip_y$	-0.4	0.3	$rhip_y$	1000	-1	1000	-0.65
$lknee_y$	1.35	0.1	$lknee_y$	300	-1.3	50	-0.55
$lankle_y$	0.35	-15	$lankle_x$	30	0	50	0
$rhip_y$	-0.4	1.0	$lankle_y$	300	3	300	-0.35
$rknee_y$	0.6	2.0	$rknee_y$	150	0.4	500	-0.5
$rankle_y$	-0.2	-9	$rankle_y$	300	-0.2	300	0.75
			$ltoe_y$	20	0	20	0
			$rtoe_y$	20	0	20	0.6

**Table 2:** *Left:* Initialization of position/angle, velocity in the start state. Unspecified DOFs are initialized to zero. *Right:* Initialization of PD control parameters. Dampers  $k_d$  are initialized to  $0.1k_p$  in all cases. Additional DOFs for both states:  $k_p = 1000, \theta_d = 0$  for  $\{rhip_{xz}, torso_{xyz}\}$ ,  $k_p = 100, \theta_d = 0$  for  $\{neck_{xyz}, rankle_x\}$ ,  $k_p = 30, \theta_d = 0$  for  $\{larm_{xz}, lelbrow_{yz}, rarm_{xz}, relbrow_{yz}\}$ , and  $k_p = 30, \theta_d = n/a$  for  $\{larm_y, rarm_y\}$ .

## References

- ALEXANDER, R. M. 2003. *Principles of Animal Locomotion*. Princeton University Press.
- ANDERSON, F. C., AND PANDY, M. G. 2001. Dynamic optimization of human walking. *J. Biomech. Eng.* 123, 5, 381–390.
- COLLINS, S., RUINA, A., TEDRAKE, R., AND WISSE, M. 2005. Efficient bipedal robots based on passive-dynamic walkers. *Science* 307, 1082–1085.
- COROS, S., BEAUDOIN, P., YIN, K., AND VAN DE PANNE, M. 2008. Synthesis of constrained walking skills. *ACM Transactions on Graphics* 27, 5 (Dec.), 113.
- DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Interactive simulation of stylized human locomotion. *ACM Transactions on Graphics* 27, 3 (Aug.), 82.
- FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *Proc. SIGGRAPH 2001*, ACM, 251–260.
- GRZESZCZUK, R., AND TERZOPOULOS, D. 1995. Automated learning of muscle-actuated locomotion through control abstraction. In *Proc. SIGGRAPH 1995*, ACM, 63–70.
- GRZESZCZUK, R., TERZOPOULOS, D., AND HINTON, G. 1998. NeuroAnimator: Fast neural network emulation and control of physics-based models. In *Proc. SIGGRAPH 1998*, ACM, 9–20.
- HANSEN, N. 2006. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms*. Springer, 75–102.
- HERR, H., AND POPOVIĆ, M. 2008. Angular momentum in human walking. *J. of Exp. Biology* 211, 467–481.
- HODGINS, J. K., AND POLLARD, N. S. 1997. Adapting simulated behaviors for new characters. In *Proc. SIGGRAPH 1997*, ACM, 153–162.
- KIM, J.-Y., PARK, I.-W., AND OH, J.-H. 2007. Walking control algorithm of biped humanoid robot on uneven and inclined floor. *J. Intelligent and Robotic Systems* 48, 4 (Apr.), 457–484.
- KUDOH, S., KOMURA, T., AND IKEUCHI, K. 2006. Stepping motion for a human-like character to maintain balance against large perturbations. In *Proc. ICRA 2006*, IEEE, 2661–2666.
- LASZLO, J., VAN DE PANNE, M., AND FIUME, E. 1996. Limit cycle control and its application to the animation of balancing and walking. In *Proc. SIGGRAPH 1996*, ACM, 155–162.
- LI, Y., WANG, W., CROMPTON, R. H., AND GUNTHER, M. M. 2001. Free vertical moments and transverse forces in human walking and their role in relation to arm-swing. *J. of Exp. Biology* 204, 47–58.
- LIU, C. K., HERTZMANN, A., AND POPOVIĆ, Z. 2005. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics* 24, 3 (July), 1062–1070.
- MACCHIETTO, A., ZORDAN, V., AND SHELTON, C. R. 2009. Momentum control for balance. *ACM Transactions on Graphics* 28, 3 (July), 80.
- MUICO, U., LEE, Y., POPOVIĆ, J., AND POPOVIĆ, Z. 2009. Contact-aware nonlinear control of dynamic characters. *ACM Transactions on Graphics* 28, 3 (July), 81.
- MUYBRIDGE, E. 1887. *Animal Locomotion*. U. of Pennsylvania.
- NOVACHEK, T. F. 1998. The biomechanics of running. *Gait and Posture* 7, 1 (Jan.), 77–95.
- POZZO, T., BERTHOZ, A., AND LEFORT, L. 1990. Head stabilization during various locomotor tasks in humans. *Exp. Brain Res.* 82, 97–106.
- SHARON, D., AND VAN DE PANNE, M. 2005. Synthesis of controllers for stylized planar bipedal walking. In *Proc. ICRA 2005*, IEEE, 2387–2392.
- SIMS, K. 1994. Evolving virtual creatures. In *Proc. SIGGRAPH 1994*, ACM, 15–22.
- SOK, K. W., KIM, M., AND LEE, J. 2007. Simulating biped behaviors from human motion data. *ACM Transactions on Graphics* 26, 3 (July), 107.
- TEDRAKE, R., ZHANG, T. W., AND SEUNG, H. S. 2004. Stochastic policy gradient reinforcement learning on a simple 3D biped. In *Proc. IROS 2004*, vol. 3, IEEE/RSJ, 2849–2854.
- TSAI, Y.-Y., LIN, W.-C., CHENG, K. B., LEE, J., AND LEE, T.-Y. in press. Real-time physics-based 3D biped character animation using an inverted pendulum model. *IEEE Trans. on Visualization and Computer Graphics*.
- VAN DE PANNE, M., AND FIUME, E. 1993. Sensor-actuator networks. In *Proc. SIGGRAPH 1993*, ACM, 335–342.
- VAN DE PANNE, M., AND LAMOURET, A. 1995. Guided optimization for balanced locomotion. In *Proc. CAS 1995*, EG, 165–177.
- WAMPLER, K., AND POPOVIĆ, Z. 2009. Optimal gait and form for animal locomotion. *ACM Transactions on Graphics* 28, 3 (July), 60.
- WITKIN, A., AND KASS, M. 1988. Spacetime constraints. *Computer Graphics (Proc. SIGGRAPH 88)* 22, 4 (Aug.), 159–168.
- WOOTEN, W. L., AND HODGINS, J. K. 1996. Simulation of human diving. *Computer Graphics Forum* 15, 1 (Feb.), 3–13.
- YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. SIMBICON: Simple biped locomotion control. *ACM Transactions on Graphics* 26, 3 (July), 105.
- YIN, K., COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2008. Continuation methods for adapting simulated skills. *ACM Transactions on Graphics* 27, 3 (Aug.), 81.