Generalized Biped Walking Control

Stelian Coros   Philippe Beaudoin   Michiel van de Panne[*]

University of British Columbia
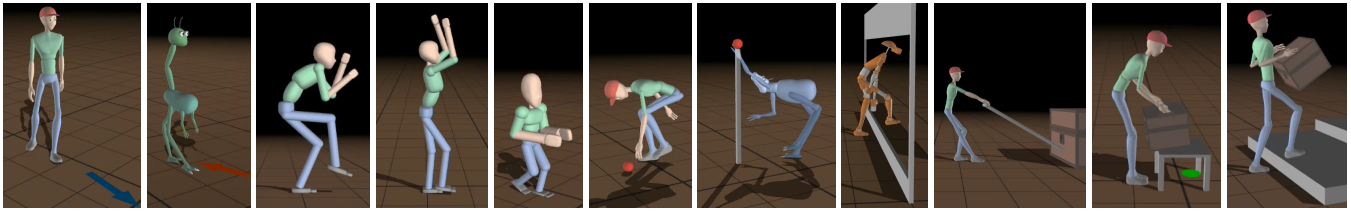
Figure 1: *Real-time physics-based simulation of walking. The method provides robust control across a range of gaits, styles, characters, and skills. Motions are easily authored by novice users.*

Abstract

We present a control strategy for physically-simulated walking motions that generalizes well across gait parameters, motion styles, character proportions, and a variety of skills. The control is real-time, requires no character-specific or motion-specific tuning, is robust to disturbances, and is simple to compute. The method works by integrating tracking, using proportional-derivative control; foot placement, using an inverted pendulum model; and adjustments for gravity and velocity errors, using Jacobian transpose control. High-level gait parameters allow for forwards-and-backwards walking, various walking speeds, turns, walk-to-stop, idling, and stop-to-walk behaviors. Character proportions and motion styles can be authored interactively, with edits resulting in the instant realization of a suitable controller. The control is further shown to generalize across a variety of walking-related skills, including picking up objects placed at any height, lifting and walking with heavy crates, pushing and pulling crates, stepping over obstacles, ducking under obstacles, and climbing steps.

1 Introduction

Physics-based character animation offers the promise of creating motion as it occurs in the world, namely as the product of muscles, gravity, and other external forces acting on a skeleton. The nuances that give a motion 'weight', such as foot impacts and subtle balance adjustments, occur as a natural byproduct when working with physical simulations. However, authoring motions in this setting is difficult, given that motions are an indirect end-product of what can be controlled, namely the forces and torques. Equally important, characters need to be balance aware if they are not to fall over. The development of good balance strategies has proven to be a vexing problem and has been the subject of several decades of ongoing research in computer animation and robotics.

A significant challenge, then, is to develop control solutions for locomotion that provide robust control over balance, that allow artists

to easily author new motions, that are fast to compute, and that generalize well in a variety of ways without requiring further tuning. Specifically, we desire generalization across *gait parameters*, *character proportions*, *motion style*, and *walking skills*. We also desire that the method should amplify an animator's authoring effort in creating interesting physics-based motions. In this context, motions should be an output of the system, and not an input. The contribution of our work is a method that addresses these issues in an integrated fashion and that can therefore be used in a very wide range of scenarios.

The proposed control strategy is inspired by a variety of previous work and consists of four key components. First, a motion generator provides target trajectories that are tracked by proportional-derivative (PD) controllers. Second, a gravity compensation model adds computed torques to the control, which allows for low-gain tracking. Third, balance-aware foot placement is achieved with the help of a predictive inverted pendulum model. Fourth, continuous balance adjustments are produced by using Jacobian transpose control to mimic the effect of a virtual desired force. When further informed by inverse kinematics for the arms and legs, the controller also allows for a variety of flexible interactions with the environment, such as being able to pick up objects or being able to step at specific locations. These components reinforce one another to achieve a novel control strategy that generalizes in a way that none of the components, taken independently, would achieve.

A wide range of results can be achieved using our method. We demonstrate the integration of walking (forwards and backwards in arbitrary user-controlled directions), standing (quiescent stance idling with foot adjustments), transitions between walking and standing, and multi-step balance recoveries from receiving a push while standing. Our characters can navigate towards a target object placed at any height and reach out to pick it up. Users can interactively create a character of desired proportions and asymmetries and immediately obtain flexible, physically simulated walking. We show that novice users and children can author novel styles with our system. The characters can also push and pull objects, climb stairs, step over obstacles, and lift-and-carry heavy crates. A caveat is that the method is best suited for dynamically-balancing but slower motions, including average speed walking. Highly dynamic motions are beyond the scope of this work.

[*]e-mail: {scoros|beaudoin|van@cs.ubc.ca}

1

2 Related Work

An expansive body of literature exists on the simulation and control of walking because of its fundamental importance to character animation, humanoid robotics, and human biomechanics. The proposed solutions often aspire to common goals and share a number of common elements.

Foot placement is a key element for achieving robust locomotion, as anyone who has been pushed can attest to. It has been at the heart of many locomotion strategies applied in animation [Raibert and Hodgins 1991; Laszlo et al. 1996; Hodgins and Pollard 1997; Yin et al. 2007; Coros et al. 2009; Wang et al. 2009] and can be traced back to its origins in robotics, e.g., [Miura and Shimoyama 1984; Raibert 1986].

Manipulation of ground reaction forces can play a key role in achieving balanced locomotion. Controllers exploiting this strategy often use full knowledge of the dynamics and current contact configuration. This knowledge allows the most desirable combination of joint torques and ground reaction forces to be computed, which are necessarily coupled and subject to a number of constraints. The desired motion can be specified using a given target trajectory to track or using other locally-defined objectives. This general class of model has been applied to animation [Abe et al. 2007; da Silva et al. 2008; Muico et al. 2009; Jain et al. 2009; Macchietto et al. 2009] and robotics, e.g., [Takenaka et al. 2009] among many others. Zero-moment point (ZMP) control strategies can also be seen as manipulating ground reaction forces in a strategic fashion.

Learning and optimization hold out the promise of developing and tuning control strategies without necessarily resorting to complex analytical models of the dynamics or careful manual tuning of parameters [Tedrake et al. 2004; Sharon and van de Panne 2005; Morimoto et al. 2007; Sok et al. 2007; Wang et al. 2009]. However, it is difficult to develop the right objective functions, and the results are usually specific to a particular character and type of motion.

Simplified models are used in some form in many locomotion controllers [Raibert 1986; Raibert and Hodgins 1991; Pratt et al. 2001; Kajita et al. 2003; da Silva et al. 2008; Yin et al. 2007; Tsai et al. 2010]. Many analytic models of control become inscrutable and intractable when applied to high dimensional systems. Simplified models, such as an inverted pendulum that approximates the motion of the body over the stance foot, provide a low dimensional system that still captures significant aspects of the dynamics.

Characters of varying body proportions can be animated by some kinematic systems, e.g., Spore [Hecker et al. 2008], and has been demonstrated to some extent for a number of dynamic systems [Hodgins and Pollard 1997; Wang et al. 2009; Tsai et al. 2010; Macchietto et al. 2009; Coros et al. 2009]. However, the goal of instantly generating natural dynamic locomotion for multiple skills applicable to arbitrary bipeds remains unsolved.

Our work shares some features of the SIMBICON control method [Yin et al. 2007] and recent related work. These methods have demonstrated the ability to perform variations in style [Yin et al. 2007], to step on or over objects [Yin et al. 2008], to walk in constrained environments [Coros et al. 2008], to integrate multiple controllers for navigation tasks [Coros et al. 2009], and can be optimized to reproduce key features of human straight line walks [Wang et al. 2009]. The controllers can still be tricky to design, however, because the balance parameters, gait parameters, motion style, and character proportions are not decoupled.

More recently, [Tsai et al. 2010] proposes the use of an inverted pendulum model (IPM) for biped character animation. Importantly, they demonstrate generalization with respect to character propor-
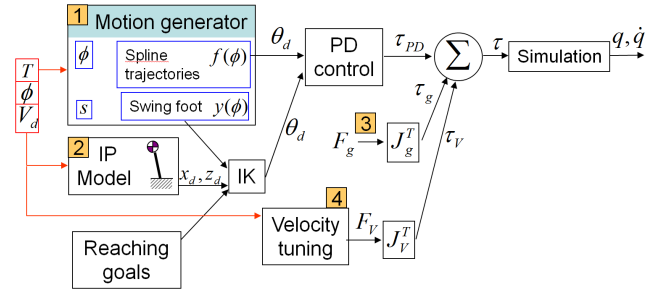

Figure 2: *System Overview. Key components of the model are: (1) a motion generator for producing desired trajectories; (2) an inverted pendulum model for predictive foot placement; (3) a gravity compensation model for all links; and (4) velocity tuning for fine balance corrections.*

tions, as this is captured by their IPM model. The approach is a *balance filter* method which tracks a motion-capture reference trajectory whose lower-body stepping motion is then modified as dictated by the IPM model.

The Jacobian transpose (JT) application of virtual forces is fundamental to force-based control in robotics. The use of JT methods to generalized control variables, such as the center of mass, was proposed in [Sunada et al. 1994] and further developed in Virtual Model Control [Pratt et al. 2001]. We use JT methods to supply computed-torque gravity compensation, as well as to apply a virtual force to the center of mass which helps regulate its velocity.

As we demonstrate in the results, each of the components of our control strategy (PD control, inverted pendulum, JT gravity compensation, and JT velocity tuning) plays an important role in achieving flexible and agile walking control. Each of these constituent components has existed in at least some basic form for 16 years. To the best of our knowledge, however, they have never been integrated to allow for the wide range of generalization, skills, and motion authoring by novice users as demonstrated in this paper.

3 Control Framework

The control system consists of four key components that are integrated as shown in Figure 2. Seen in broad terms, these components work as follows. The motion generator produces open-loop desired joint angle trajectories, which are modeled as spline functions over time. Joint angle trajectories can be modeled relative to their parent-link coordinate frame or relative to the character coordinate frame (CCF), whose y and x-axes are aligned with the world 'up' vector and the character's facing direction, respectively. The use of CCF-relative target joint angles introduces additional feedback into the system because of the implicit knowledge at these joints of which way is up.

A robust balance mechanism is added through the use of foot placement, which is computed with the help of an inverted pendulum model (IPM). The IPM helps achieve motion that is highly robust to disturbances such as pushes. As noted in [Tsai et al. 2010], a feature of IPM-based foot placement is that it does not require parameter tuning because the relevant parameters are captured by the model.

We augment the PD-control with computed-torque gravity compensation. This compensation provides a simple-to-compute estimate of a significant component of the required control, while avoiding the complexity of needing to invert the dynamics of the motion. The

addition of virtual gravity compensation forces, F_g, and the torques that implement them, τ_g, allows for acceptable accuracy tracking to be achieved using low-gain PD-tracking in joint space. The low gains, in turn, help allow for natural, highly compliant motion. The virtual gravity compensation forces are computed for each link and are referred back to the root link, namely the pelvis.

We implement fine-scale control by applying a virtual force, F_V, on the center of mass in order to accelerate or decelerate it towards the desired velocity. Like the gravity compensation forces, the torques to implement this are computed using the Jacobian transpose. We now describe each of the components in further detail.

3.1 Motion Generator

The *motion generator* produces the various desired trajectories that help create desired motion styles. The trajectories directly model desired joint angles, either relative to their parent joints (elbows, shoulders, stance knee, and toes) or relative to the character coordinate frame (waist, back, neck, and ankles). The character frame is defined by axes aligned with the vertical axis of the world and the facing direction of the character. The desired joint angles are provided as input to PD-controllers that produce tracking torques. The trajectories are modeled as a function of the phase of a step, $\phi \in [0, 1)$ using Catmull-Rom splines. A walk cycle then consists of two steps, with the second step being left-right symmetric to the first. The number of spline segments is arbitrary, although we typically use three segments. The final simulated motions do not necessarily tightly track the target trajectories and contain significant details that are not present in the desired trajectories. The crate lifting and crate pulling are good examples of this (Section 5.4).

The motion generator trajectories can be authored using keyframes, as described in Section 5.2. In general, a controller will work robustly with all angle trajectories being set to zero, which implies having the arms pointing straight down, a straight stance leg, feet and toes remaining parallel to the ground, and a fully upright body. This 'zero gait' provides a logical starting point for end-user authoring of gaits. When not zero, the target trajectory for the stance knee is assigned a constant value in order to achieve a bent-leg crouched gait, for example. The stance ankle trajectory can be driven to achieve tip-toe walks or other desired toe-off patterns. As will be described shortly, the stance hip and the swing leg are controlled separately and thus these will still actively produce joint motion in the case of all zero joint angle trajectories. Only the $y(\phi)$ curve, which specifies the swing foot height, need be non-zero (Section 3.2).

To allow for simple and coordinated control over bending of the body, the target angles for the lower back, upper back, and head are modeled as fixed ratios of a single user controllable bend angle in the sagittal plane. The constraint can be dropped in order to allow for more artist control and it is not enforced in our style editing interface (Section 5.2).

Given a desired time period for a single step, T, the current phase of an ongoing motion is computed as $\phi = t/T$. The motion generator keeps track of the current stance leg using a two-valued state variable, $s \in \{left, right\}$. A new step is assumed to begin at foot-strike or when $t \geq T$, whichever occurs first. Following [Yin et al. 2007], our motion generator does not provide target trajectories for the stance hip. Instead, its torque is computed to achieve the desired net torque on the pelvis, which is the root link of the character.

3.2 Inverted Pendulum Foot Placement

Computing where to step and how to achieve that step is an important problem for walking skills. The steps we wish our controller to perform can vary widely in nature, ranging from making minimal shifts of foot position during idle stance to taking large steps during a fast walk or when receiving a large push.

We compute the desired stepping point, (x_d, z_d), using an inverted pendulum model (IPM) in order to achieve a general solution that works across a wide range of body types. In particular, we build on an inverted pendulum model that assumes constant leg length [Pratt and Tedrake 2006] as shown in Figure 4 (left). The analysis equates the sum of the potential and kinetic energy of the IPM at the current state, described by its current velocity v, its height, h, and distance, d, from the future point of support, with that at the balanced rest state, i.e., $\frac{1}{2}mv^2 + mgh = \frac{1}{2}mv'^2 + mgh'$, where $v' = 0$ and $h' = L = \sqrt{h^2 + d^2}$. Solving this relation for d gives $d = v\sqrt{h/g + v^2/(4g^2)}$. The above model computes the desired value of d in order to reach zero velocity at the next step. Taking a shorter step will achieve a positive velocity while taking a larger step will achieve a negative velocity, i.e., walking backwards. Accordingly, we compute $d' = d - \alpha V_d$, where V_d is the magnitude of the desired velocity and α is a constant. We use $\alpha = 0.05$ for all the results demonstrated in this paper. Because the velocity is also controlled using the velocity tuning component, the control is not particularly sensitive to the particular value of α (Section 6). We have experimented with other variations of the IPM, including one that directly takes into account a non-zero desired velocity. Our final choice of IPM as described above is based on the overall robustness and quality of motion. The IPM is applied in the sagittal plane to compute $x_d = d$ and is repeated in the coronal plane to obtain z_d in an analogous fashion. We use the true center of mass position and velocity when applying the pendulum model to the biped. The target values x_d and z_d are updated at every time step, although if a particular foot placement in the world is desired, the character can ignore the IPM prediction for one or two consecutive steps. Depending on the speed of the character, the IPM prediction may be out of reach, in which case the characters uses a maximum step length of $d = 0.6L$. This situation leads to the character possibly having to take multiple steps to recover.

We drive the motion of the swing leg by synthesizing a desired trajectory for the swing ankle relative to the ground and in the character coordinate frame. The height with respect to the ground, y, is modeled as a function of phase by a three-segment spline curve in the motion generator, i.e., $y(\phi)$. This curve defines the difference between a ground-hugging step and a high leg-lift step. The x and z trajectories follow linear trajectories between their locations at the start of the step (x_0, z_0) and their desired location at the end of the step, (x_d, z_d), i.e., $x(\phi) = (1 - \phi)x_0 + \phi x_d$ and $z(\phi) = (1 - \phi)z_0 + \phi z_d$. Inverse kinematics is used to compute target joint angles for the swing hip and knee, which are then tracked using PD controllers and augmented by gravity compensation torques. The inverse kinematics problem has a remaining degree of freedom which allows for knock-kneed, normal, or bowlegged walking variations. We expose this as a twist angle parameter to the animator which is held constant within any given style of motion.

3.3 Velocity Tuning

While foot placement ensures robustness for the gait, it can only be enacted once per step. Using foot placement alone ignores the ability to use the stance foot (or feet) to help maintain balance by manipulating the ground reaction forces (GRFs), or equivalently, manipulating the location of the center of pressure (COP), also known
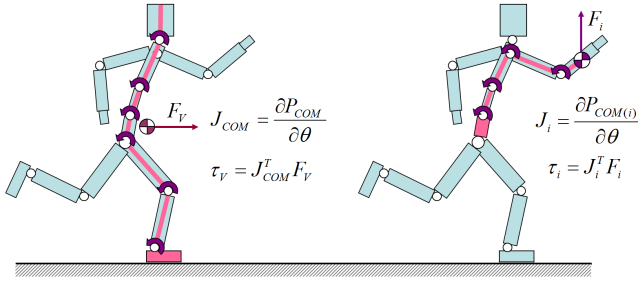
Figure 3: *Jacobians used for velocity tuning (left) and gravity compensation (right).*

as the zero-moment point (ZMP). Simply put, shifting the COP towards the toes helps in slowing the forward progression of the body, while shifting it back helps accelerate. Many biped walking control algorithms use some form of this manipulation as their primary control strategy. However, this then requires the precomputation of feasible reference trajectories and often assumes full knowledge of the dynamics. In our walking controller, we use a simple virtual velocity-tuning force, akin to that proposed in [Pratt et al. 2001], to provide fine-scale control over the center of mass velocity. It has the effect of altering the GRFs and the COP and so we place this control strategy in the same general category as other techniques that manipulate the GRFs, COP, or ZMP in order to enact balance feedback.

The specific details of computing and applying the virtual force, F_V, on the center of mass are as follows. We first compute the center of mass velocity of the biped, V. In the sagittal plane, we compute a desired virtual force according to $F_V = k_V(V_d - V)$, where V_d is the desired sagittal velocity. In the coronal plane, an analogous virtual force is computed using a PD-controller that tracks a desired lateral COM position. This desired position varies linearly over time from its original position at footstrike to a step width W at the time of the next predicted footstrike. By default, using $W = 0$ results in a catwalk style of motion. A non-zero value of W results in the stance leg pushing the body to the side, which then causes the inverted pendulum model to compensate accordingly with a lateral swing-foot placement.

The combined sagittal-and-lateral virtual force is achieved using $\tau_V = J_V^T F_V$. Here, J_V is the Jacobian of the center of mass for a chain of links that is considered rooted at the stance ankle. In practice, we compute J_V using only the joints between the stance foot and the head, as shown in Figure 3 (left), which incorporates all the key joints that support most of the weight. The virtual force helps fine tune both sagittal and lateral balance. Because of the finite support of the feet, applying τ_V can in some situations cause undesired foot rotation, e.g., the toe rotating off the ground in an attempt to accelerate forwards. To mitigate this problem we zero the ankle component of the torque if we detect that the stance foot is not well planted, as measured by the toe leaving the ground or the foot exceeding a rotational velocity threshold. There is potential to have 'chatter' occur, but in practice we have not found this to be the case. During double stance each stance leg is treated independently from its foot to the torso and the results are simply summed.

3.4 Gravity Compensation

The use of computed gravity compensation (GC) torques allows for the use of significantly lower gains in the PD controllers for the limbs and the body. Although highly controllable motion could

also be achieved with the use of a full inverse dynamics solution, we show that this is unnecessary. The addition of gravity compensation allows simple local control methods, such as low-gain PD-control, to succeed.

Gravity compensation is applied using a Jacobian transpose method once again. We wish to apply a virtual force $F_i = -m_i g$ at the center of mass of every link i, where the negative sign implies an upwards force. The torques required for this are computed as $\tau_i = J_i^T F_i$. Here, J_i is the Jacobian of the link center of mass location with respect to the chain of joints between the root link and link i. GC torques are thus computed for all joints that lie in between each link i and the root link, i.e., the pelvis. Any given joint j thus sees the sum of the GC torques required by all links that are distal to it. The compensation is applied to all links, with the exception of those in the stance leg.

3.5 Turning and Limb Guidance

Turning is achieved using the stance hip, similar to [Yin et al. 2007], with a desired turning rate of $\omega_{max} = 2\,\mathrm{rad/s}$. A turning phase, $\phi_{turn} \in [0, 1]$, is defined as the fraction of the turn completed. The desired COM velocity, V_d, is interpolated between its value before the turn and after the turn using ϕ_{turn}. The stance hip twist angle tracks a target facing direction that is interpolated using ϕ_{turn}. The lower-back, upper-back, and head each rotate to anticipate the turn according to linearly weighted functions of ϕ_{turn} so that the head leads the anticipatory movement. The swing-leg IK plane of rotation also anticipates the turn, which helps plant the swing foot in an anticipatory fashion.

During sharp turns, the default trajectory for the swing leg may cause it to collide with (or in our case, pass through) the stance leg. If the swing foot trajectory passes on the wrong side of the stance foot, a 3-segment spline curve is computed based upon the current position, the target position, and a fixed waypoint. This strategy greatly reduces the number of self collisions, although in our implementation it does not completely eliminate them. Analysis of human swing leg motions is a possible principled alternative for dealing with this issue, but it is unclear if it would generalize well across gaits, styles, and character proportions.

Cartesian-space trajectories are used to guide the motion of the swing foot, as presented earlier. Given the current world-coordinates target position for the swing foot, inverse kinematics is used to compute the target angles for the swing hip and knee. With this approach, a bent stance knee automatically results in a matching bent swing leg motion since the hips are lower. This approach also allows the controller to guide the swing foot over an obstacle by only changing the desired foot height. If needed, the foot target location computed by the inverted pendulum model can be ignored in favor of precisely positioning the foot at a specific ground location. However, this typically cannot be done for multiple steps in a row since it ignores balance considerations.

Physics-based characters need to be equipped with skills that go beyond basic walking if they are to be used extensively in games and simulations. One fundamental skill is that of being able to reach towards specific points in the environment in order to pickup an object, place a hand on a door handle, or to turn on a light switch. In our framework, the desired joint angles for the shoulder and elbow are computed with inverse kinematics once the character is within reaching distance. Gravity compensation is key to achieving successful motions without resorting to excessively high PD gains.

We employ an analytic solution to the two-link inverse kinematic problems [Kulpa et al. 2005]. To compute a unique solution, the elbow or knee is forced to lie in a plane that also embeds the shoul-
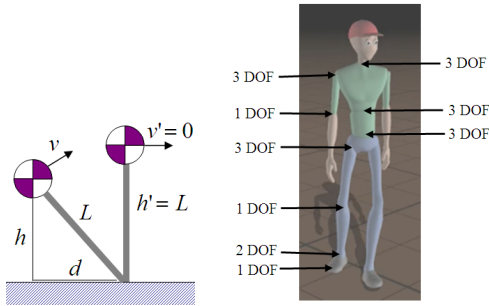
Figure 4: *Left: Inverted pendulum model. Right: Character degrees of freedom.*

der and hand, or analogously for the leg, the hip and ankle. The rotational degree of freedom of this embedding plane is specified as an input parameter that allows for bow-legged walks or control over the style of arm reaching movements.

4 Implementation

A version of our implementation will be made available as an open source project. We use *Open Dynamics Engine* [ODE] as our forward dynamics simulator. A simulation time step of $0.0005\,s$ is used. All balance control parameters are kept fixed across all our simulations, i.e., there are no parameters to tune by a user interacting with the system. For PD-gains, we use k_p values that scale with the total mass of the character being controlled. Specifically, when an edit creates a new character with mass M', we scale the k_p values by M'/M, where M is the original character mass. We use $k_d = 2 * \sqrt{k_p}$ for all joints. Without PD scaling, the first point of failure is typically for large characters whose knees begin to buckle under their weight. The coefficient of friction between the character and the ground is 0.8. We do not use torque limits in the specific results we show. However, we have experimented with adding torque limits ($|\tau| \leq 200Nm$, for hips and knees, $|\tau| \leq 100Nm$ for all other joints) for our humanoid character and this did not have any visible impact on the motions. With the exception of a simulated crowd scene, all the simulations run in real time, written as a single-threaded application and run on a 2.4GHz 2-core machine with 2Gb of memory. We do not process collisions between body parts belonging to the same character.

Our characters have the degrees of freedom shown in Figure 4. There are a total of 37 DOF, which includes 6 DOF for the global position and orientation. Our *humanoid* character (Figure 5, row 1) has dimensions and a mass distribution that are typical of a human. The *robot* character (Figure 5, row 4) has similar dimensions, though its left arm is 40% longer and heavier, and the left knee is 20 cm higher than the right knee. The *beast1* and *beast2* characters (Figure 5, rows 2 and 3) have much of their mass in the link that corresponds to their pelvis. Characters created using the character editor have link masses that are proportional to their volume, i.e., the density is assumed to be constant.

5 Results

We demonstrate a variety of animation results. These are best seen in two videos: the accompanying video and the supplementary material video.

5.1 Generalization across gait parameters

We first show that the walking control generalizes across forwards-and-backwards walking, walking speeds, stepping frequencies, and turning towards a desired direction. We also show integrated stopping and starting behaviors. We demonstrate these basic behaviors across different characters and for different styles of motion. Figure 5 shows frames from the animations. A commanded forward walk is indicated by a blue arrow, a backwards walk by a red arrow, and a green dot indicates a command to stop walking.

The desired speed, V_d, is given as an input to the control system, as shown in Figure 2, and is achieved using a combination of foot placement (via the inverted pendulum model) and Jacobian transpose control (velocity tuning). The desired time period, T, is an input to the motion generator, where it is used in computing the current phase, $\phi = t/T$. Walking with time periods of $0.2\,s \leq T \leq 1.0\,s$ works well for characters of approximately human scale. Very long time periods ($T > 1s$) with medium-to-large steps are problematic as the character needs to pause when over the stance foot, while our control strategy assumes a constant desired speed. The average stepping length, L, is implicitly determined by $L = V_dT$. Setting $V_d < 0$ results in backwards walking. Our characters walk at speeds ranging from -0.6 to $1.7m/s$.

When the character is stopped, balance requires no special treatment – it is achieved using the velocity tuning component, with the torques being referred to both the right and left feet. The sagittal and lateral components of the virtual force F_V are both computed using a PD-controller that now tracks a static target position for the COM that is located at the midpoint between the two feet. To begin walking, the COM-target is simply switched to lie in the middle of the desired future stance foot. Once the COM has moved 40% of the way, walking is triggered. This trigger event is equivalent to a footstrike in the walking model, i.e., it represents the initiation of a new step. To stop walking, V_d is set to zero. Once $V \leq \epsilon$ is achieved at footstrike, which may require more than one step, the standing balance control is invoked as described earlier. An idling behavior is used to create quiescent stance motions that shift the weight between the legs and to make small adjustments of the un-weighted foot. The behavior is scripted using a simple finite state machine.

The character can also recover from pushes while standing by invoking walking when needed, using $V_d = 0$ and the same stop-walking condition described above. The COM position and velocity is monitored while standing. If the COM position is too close to the edge of the support polygon or $V > \epsilon$, then walking is invoked in order to regain balance by stepping.

We demonstrate control over gait parameters for characters of varying proportions and for different styles in order to show generalization across the cross-product of these features. Figure 5 illustrate the execution of a sequence of commanded gait parameters over four characters of varying proportions and dimensions. No character-specific or style-specific parameter tuning is required.

5.2 Generalization across style

The controller produces balanced motions for a wide variety of user-authored styles of motions. Rows 1, 5, and 6 in Figure 5 illustrate three different walking styles for the same character. These styles were authored by modifying the lean of the upper body in the sagittal plane, the target angle of the stance knee, the plane of rotation for the swing-leg IK, and the arm motions.

We also develop an interface that allows animators to easily author unique motion styles. Figure 6 shows the interface and example
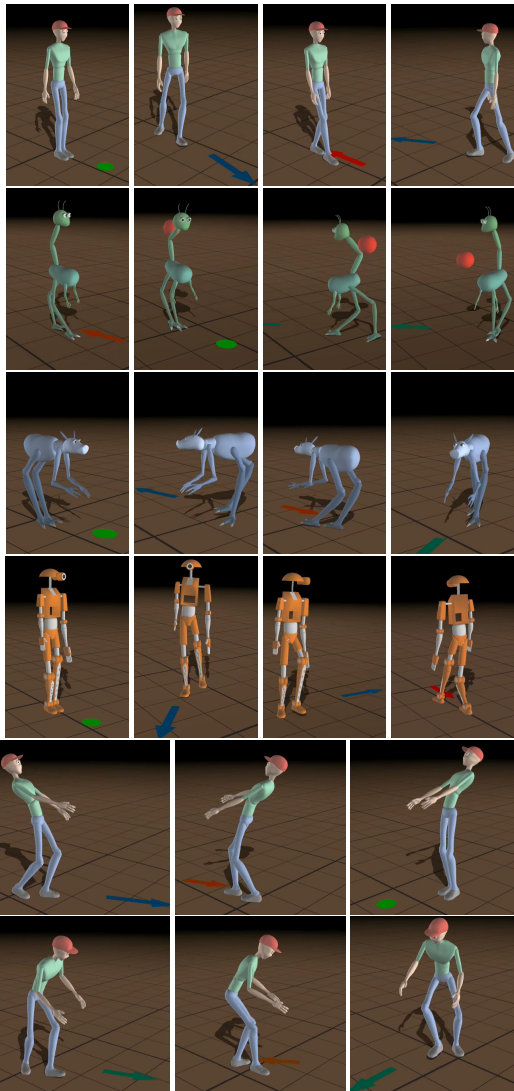
Figure 5: *Direction Following*

Figure 6: *Motion style editing and example motion styles.*
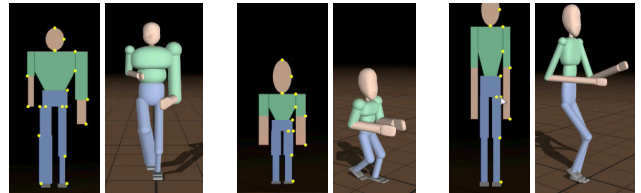
Figure 7: *Interactive editing of character proportions and the resulting walking gait.*

motion styles that were created in 5-10 minutes by novice users, including children as young as 7. Our walking control strategy instantly provides robust dynamically-simulated gaits. The interface is constrained to symmetric trajectories for the sake of simplicity.

5.3 Generalization across characters

The control strategy generalizes well across a variety of character dimensions and proportions, as shown in rows 1–4 of Figure 5. We further develop an interface that allows a user to interactively edit the dimensions and proportions of a character and immediately see the resulting motions. The interface and examples of resulting characters are shown in Figure 7. Characters can have asymmetric arms and legs, which will introduce asymmetries in the motion even with symmetric controls from the motion generator. Asymmetry in the placement of the knee joint, such as for our robot character, is largely accommodated by the inverse kinematics used for tracking the desired swing ankle trajectories.
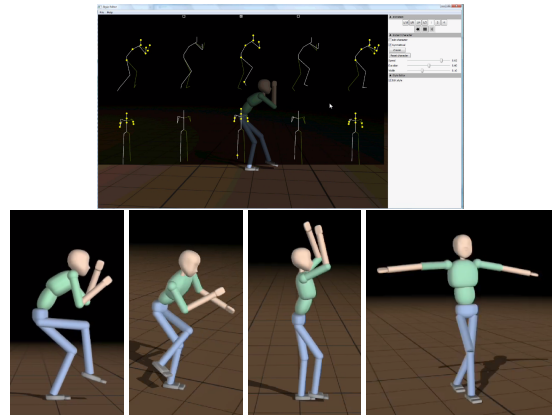
5.4 Generalization across tasks

Walking controllers should also generalize across tasks and be able to interact with the environment in various ways. We demonstrate this on several different tasks.

Reaching: Figure 8 shows the ability to reach and grasp an object placed at an arbitrary location and at an arbitrary height. We demonstrate object reaching motions on three different characters, showing generalization across character proportions. The reaching strategy uses a high-level behavior that controls the walking direction in order to guide the character towards the target object. A desired heading direction is computed once per step for the character based on the location of the object. Most reaching motions can be carried out while walking at a regular pace. Objects near the ground require the character to slow down, which is implemented as a linear function of the remaining distance, d, to the object. Low reaches also require bending down, using a combination of forward upper body lean and bending of the stance knee. The reaching motion is triggered when $d < \epsilon$, at which point the arm treats the final object position as a target. Inverse kinematics provides required joint angles, which are then tracked using low-gain PD-controllers and augmented by the gravity-compensation torques.

Pulling and pushing a crate: The walking control can be used to pull and push objects, as shown in Figure 9. The crate weighs $80 \, \text{kg}$ and has a coefficient of friction of 0.2 with the ground. The high level behavior has the character walk towards the crate, slow down when near the crate handles, and reach for the handles when within reach. Springs and dampers are used to connect the hands to the handles. The hands are not animated in our implementation and are treated as an extension of the lower arm. The user can interactively control the target bend parameter of the body to produce sporadic tugging behavior, if desired. The inverted pendulum model is adapted to account for the weight to be pushed or pulled by using $\alpha = 0.2$ instead of $\alpha = 0.05$.
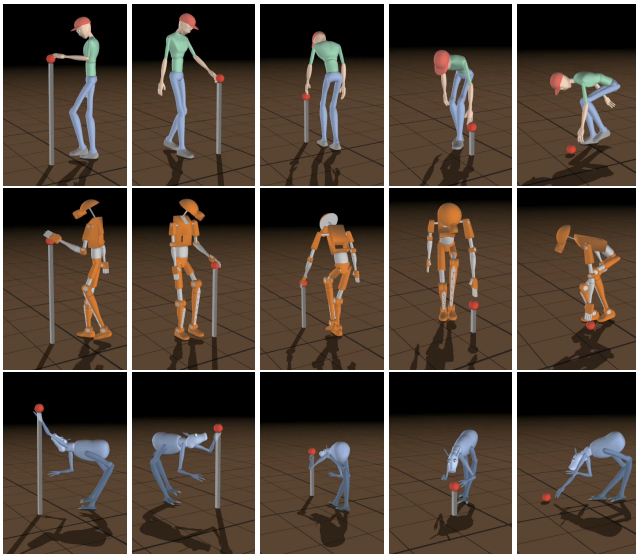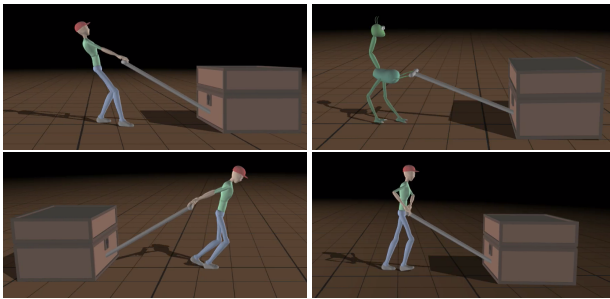
Figure 8: *Reaching for objects*

Figure 9: *Pulling and pushing a crate. Top: Pulling to the left. Bottom: Pulling and pushing to the right.*

Figure 10: *Lifting and moving heavy crates.*

Figure 11: *Navigating over and under obstacles. Top: Stepping over a sequence of obstacles of up to 45cm in height. Bottom: Stepping over a sequence of obstacles combined with ducking under obstacles, with varying offsets.*

Lifting and moving a crate: The simulated characters can lift and move heavy crates (5–25 kg) using the generalized walking control, as shown in Figure 10. The high level behavior used to approach the crate is similar to that used in the pulling and pushing skill. Once the hands are attached to the box with stiff springs and dampers, the crate mass is incorporated into the computations for the overall COM position and velocity. It is also incorporated into the gravity compensation by adding a compensation force corresponding to half the weight of the box to each hand. The hands are not animated. The user controls the walking speed and direction interactively once the character has lifted the crate.

Navigation over and under obstacles: A high level behavior that steps over obstacles and ducks under other obstacles is also easy to create. Of particular interest is that the stepping-over behavior and the ducking behavior are controlled independently as they do not interfere with each other. Figure 11 shows results for two different humanoid characters in navigating an obstacle course. The step-over and duck-under obstacles are paired together with varying offsets in order to demonstrate that the factored control works well for all combinations of these two skills. The control for ducking under obstacles consists of bending the body as a linear function of the obstacle distance, yielding a smooth ducking motion.

The high level control for stepping over an obstacle involves slowing down upon approaching the obstacle. It is then important to arrang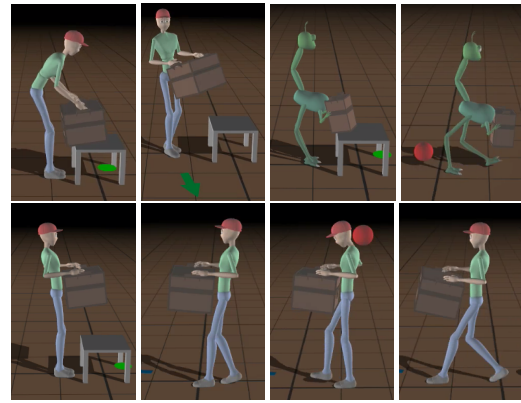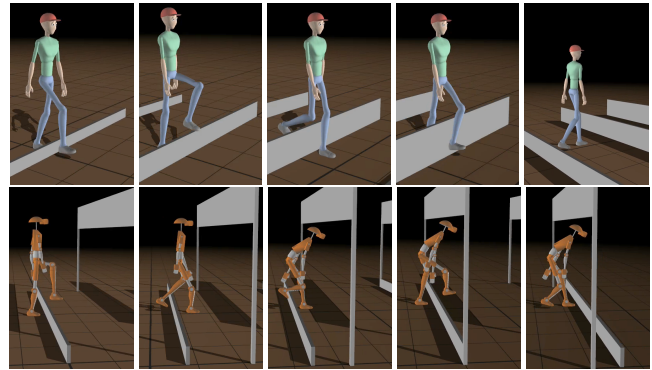e for a footstep that is planted near the obstacle itself before stepping over it. This footstep is done by adapting the step length once within a threshold distance of the obstacle. Because the location of the swing foot is determined by the inverted pendulum model, we implicitly adjust the step length by manipulating the desired velocity. Once in position to step over the obstacle, a trajectory is planned that allows the swing foot to clear the obstacle in the upcoming step. This trajectory guides the swing ankle and is constructed using 4 Catmull-Rom Spline segments. The swing foot follows this trajectory until the heel is past the obstacle, after which control of the forward component once again reverts to the inverted pendulum in order to produce a final foot placement that is suitable from the point of view of balance. The motion of the trailing foot over the obstacle is controlled in an analogous fashion.

Stairs: Figure 12 shows the humanoid character carrying a 15 kg crate as it walks up 15 cm stairs and steps over 20 cm obstacles. The high level control for climbing a step is identical to that for stepping over an obstacle, except for the way in which the vertical trajectory of the swing ankle is computed.

Crowd Simulation We simulate a crowd of 16 dancing characters of various sizes, shown in Figure 13. The characters occasionally bump into each other or step on each other's feet, which creates a scene that is rich in motion detail. Creating this scene simply requires instancing 16 characters with randomized proportions and gait parameters and no other special effort. The simulation runs approximately 10-times slower than real time. The simulation speed varies as the number of contacts increases and decreases.
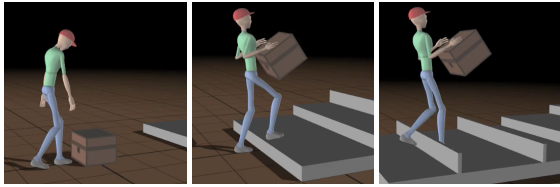
Figure 12: *Walking to a crate, picking it up, climbing steps, stepping over obstacles, and coming to a stop.*

Figure 13: *A dancing crowd with 16 characters.*

6 Analysis and Discussion

The use of the suitable control representations and simplified models allows for significant generalization to occur across body types and motion styles without needing to resort to learning or needing to invert the full dynamics.

Representations and generalization: An important test of the utility of a motion control model is to see how well it generalizes across a range of motions and characters. In our results, we demonstrate a significant degree of generalization across gaits, styles, characters, and skills. The control strategy achieves much of its power from the way the control is factored into several components which can then be composed together. The individual components each see only a limited aspect of the overall dynamics.

A viable alternative approach is to assume that the dynamics are fully known and to exploit this in the control strategy [da Silva et al. 2008; Macchietto et al. 2009; Jain et al. 2009]. In particular, a model of the inverse dynamics allows a target trajectory to be tracked, and in this way motions can be represented directly by desired motion trajectories. The presence of underactuation and contacts make this a challenging problem for walking, but with appropriate prediction these can also be taken into account [Muico et al. 2009]. One way to achieve generalized control within a trajectory-tracking model might be to parameterize the input physically-feasible target motion trajectories with respect to gait, style, character dimensions, and skills. However, this is a challenging problem to solve and it resembles the desired output of an animation system rather than an input.

Our proposed control strategy is constructed around approximate models that can be used in support of executing new motions with a reasonable degree of skill. It lends some support to the hypothesis that, with the right representation, *"Bipedal walking is an inherently robust problem, not requiring precision, accuracy, or repeatability"* [Pratt and Drakunov 2007]. The computational requirements of the control are low as no implicit or explicit inversion of the system dynamics that is required. The method does not need online or offline optimizations.

Anticipation: Skilled motion generally requires a strong degree of anticipation of the forces required to perform a motion. There are several anticipatory elements in our control strategy. The inverted

pendulum is a simplified dynamics model that makes specific predictions regarding foot placement to achieve a desired velocity. The gravity compensation model anticipates a significant degree of the control required for low-speed motions. The PD-controllers only need to deal with inertial dynamics. The velocity tuning has knowledge of the mass distribution of the body and its kinematics in relation to the point of support. Taken in combination, these control elements exhibit a significant degree of anticipation. To achieve further anticipatory control, we expect that the PD control could be largely replaced by feed-forward torques that are learned [Yin et al. 2007] or that are computed from a reference trajectory using inverse dynamics.

Authoring: Computer animation tools need to provide artist-friendly representations. As shown in the accompanying video, we have developed a simple interface to support interactive editing of motion styles and character proportions. Additional parameters, such as the turning rate can also be exposed as high level parameters. angles of the embedding to achieve knock-kneed or bow-legged With these tools, users with no prior animation experience are able to create highly varied characters and motion styles in 5–10 minutes during unconstrained exploration, i.e., we do not ask them to reproduce a specific reference motion style. The rules that encode high level skills such as "go pick up the crate and carry it to here" are currently scripted.

Robustness: We test the robustness of the humanoid character by applying external disturbance forces to the torso for a duration of $0.1\,\mathrm{s}$. During a forward walk of $0.6\,\mathrm{m/s}$, the character is robust to omni directional pushes of $600\,\mathrm{N}$, applied at arbitrary points in the walk cycle. The mass of the character is $70.4kg$.

Parameter settings: We use fixed parameter settings across all motions, unless otherwise noted. These include $\alpha = 0.05$, the inverted pendulum velocity gain; $k_v = 100Ns/m$, the velocity tuning gain; and $\omega_{max} = 2\,\mathrm{rad/s}$, the maximum turning rate. A default trajectory needs to be provided for the height of the swing ankle as a function of phase, and a self-collision avoidance strategy needs to be designed that helps avoid the swing leg colliding with the stance leg during sharp turns.

In the supplementary material video, we demonstrate the effect of doubling and halving the PD-gains used for joint tracking. The overall motion remains similar, although high-gain tracking results in a stiffer-looking motion, while the very low-gain tracking gives a slightly sloppy and weak looking motion. We have also examined the effect of changing α, the parameter that modifies the prediction of the inverted pendulum mode. A value of 0 works well for desired speeds $|V_D| \leq 0.4$. For larger V_D values, the walk cycle can become asymmetric, as more energy is lost due to the location of the step than can be injected during one step by the virtual model control. We have also tried doubling the value of α, without any noticeable decrease in robustness. However, as the value of α keeps increasing, the predicted foot location may become unsuitable for maintaining balance.

Contributions of each control component: Each of the core components plays an important role in achieving robust and flexible control. In the supplementary material video, we show the effect of removing individual control components. The effect of removing gravity compensation results in a slouched upper body, unsuccessful reaching motions, and less accurate placement of the swing foot. Removing the inverted pendulum control results in an inability to recover from any significant perturbations. Without velocity tuning the velocity is controlled with much less precision and the characters have difficulty walking at steady state for a given velocity.

Limitations: While the current framework allows authoring a motion style, it does not allow for the authoring of a 'push recov-

ery style', as this aspect of the motion is currently governed by the inverted pendulum foot-placement. Self-intersection between the swing and stance legs still occurs in some situations. We do not demonstrate generalization across terrain, with the exception of climbing steps. We have not explored extending the method to non-biped morphologies. The motions that we produce are agile in the sense of being able to achieve fast turning behavior and fast transitioning between forwards and backwards walking. However, we have not demonstrated high-speed agility. Motions which are particularly demanding with respect to energy or other performance measures may still be challenging to author and would benefit from optimization.

7   Conclusions

Developing the control required for physics-based skills is a significant challenge. We have presented a simple and general method for biped walking control. The method generalizes across gait parameters, motion styles, character proportions, and locomotion tasks. We demonstrate that naive users can interactively author the character proportions, gait parameters, and motion styles. We hope that the technique may enable the wider adoption of physics-based characters in games and film.

References

ABE, Y., DA SILVA, M., AND POPOVIĆ, J. 2007. Multiobjective control with frictional contacts. In *Proc. ACM SIGGRAPH/EG Symposium on Computer Animation*, 249–258.

COROS, S., BEAUDOIN, P., YIN, K., AND VAN DE PANNE, M. 2008. Synthesis of constrained walking skills. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA) 27*, 5, Article 113.

COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2009. Robust task-based control policies for physics-based characters. *ACM Trans. on Graphics (Proc. SIGGRAPH ASIA) 28*, 5, Article 170.

DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Interactive simulation of stylized human locomotion. *ACM Trans. on Graphics (Proc. SIGGRAPH) 27*, 3, Article 82.

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 (Proc. SIGGRAPH) 27*, 3.

HODGINS, J. K., AND POLLARD, N. S. 1997. Adapting simulated behaviors for new characters. In *Proc. ACM SIGGRAPH*, 153–162.

JAIN, S., YE, Y., AND LIU, C. K. 2009. Optimization-based interactive motion synthesis. *ACM Trans. on Graphics 28*, 1, 1–10.

KAJITA, S., KANEHIRO, F., KANEKO, K., FUJIWARA, K., HARADA, K., YOKOI, K., AND HIRUKAWA, H. 2003. Biped walking pattern generation by using preview control of zero-moment point. In *Proc. IEEE Int'l Conf. on Robotics and Automation*.

KULPA, R., MULTON, F., AND ARNALDI, B. 2005. Morphology-independent representation of motions for interactive human-like animation. In *Computer Graphics Forum*, vol. 24, 343–352.

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

MACCHIETTO, A., ZORDAN, V., AND SHELTON, C. R. 2009. Momentum control for balance. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3.

MIURA, H., AND SHIMOYAMA, I. 1984. Dynamic walk of a biped. *Int'l J. of Robotics Research 3*, 2.

MORIMOTO, J., ATKESON, C. G., ENDO, G., AND CHENG, G. 2007. Improving humanoid locomotive performance with learnt approximated dynamics via guassian processes for regression. In *Proc. IEEE Int'l Conf. on Robotics and Automation*.

MUICO, U., LEE, Y., POPOVIC', J., AND POPOVIC', Z. 2009. Contact-aware nonlinear control of dynamic characters. *ACM Trans. on Graphics (Proc. SIGGRAPH) 28*, 3, Article 81.

ODE. Open dynamics engine, `http://www.ode.org/`.

PRATT, J. E., AND DRAKUNOV, S. V. 2007. Derivation and application of a conserved orbital energy for the inverted pendulum bipedal walking model. In *Proc. IEEE Int'l Conf. on Robotics and Automation*.

PRATT, J. E., AND TEDRAKE, R. 2006. Velocity based stability margins for fast bipedal walking. In *Fast Motions in Biomechanics and Robots*.

PRATT, J., CHEW, C., TORRES, A., DILWORTH, P., AND PRATT, G. 2001. Virtual model control: An intuitive approach for bipedal locomotion. *Int'l J. Robotics Research 20*, 2, 129.

RAIBERT, M. H., AND HODGINS, J. K. 1991. Animation of dynamic legged locomotion. In *Proc. ACM SIGGRAPH*, 349–358.

RAIBERT, M. H. 1986. *Legged Robots That Balance*. MIT Press.

SHARON, D., AND VAN DE PANNE, M. 2005. Synthesis of controllers for stylized planar bipedal walking. In *Proc. IEEE Int'l Conf. on Robotics and Automation*.

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

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

TAKENAKA, T., MATSUMOTO, T., AND YOSHIIKE, T. 2009. Real time motion generation and control for biped robot, first report: Walking gait pattern generation. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*.

TEDRAKE, R., ZHANG, T., AND SEUNG, H. 2004. Stochastic policy gradient reinforcement learning on a simple 3D biped. In *Proc. Int'l Conf. on Intelligent Robots and Systems*, vol. 3.

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. IEEE Trans. on Visualization and Computer Graphics.

WANG, J., FLEET, D. J., AND HERTZMANN, A. 2009. Optimizing walking controllers. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*.

YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. SIMBICON: Simple biped locomotion control. *ACM Trans. on Graphics (Proc. SIGGRAPH) 26*, 3, Article 105.

YIN, K., COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2008. Continuation methods for adapting simulated skills. *ACM Transactions Graph. (Proc. SIGGRAPH) 27*, 3.