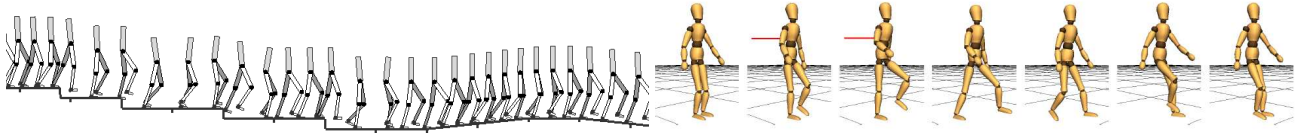


# SIMBICON: Simple Biped Locomotion Control

KangKang Yin    Kevin Loken    Michiel van de Panne\*

University of British Columbia



**Figure 1:** Real-time physics-based character simulation with our framework. (a) A single controller for a planar biped responds to unanticipated changes in terrain. (b) A walk controller reconstructed from motion capture data responds to a 350N, 0.2s diagonal push to the torso.

## Abstract

Physics-based simulation and control of biped locomotion is difficult because bipeds are unstable, underactuated, high-dimensional dynamical systems. We develop a simple control strategy that can be used to generate a large variety of gaits and styles in real-time, including walking in all directions (forwards, backwards, sideways, turning), running, skipping, and hopping. Controllers can be authored using a small number of parameters, or their construction can be informed by motion capture data. The controllers are applied to 2D and 3D physically-simulated character models. Their robustness is demonstrated with respect to pushes in all directions, unexpected steps and slopes, and unexpected variations in kinematic and dynamic parameters. Direct transitions between controllers are demonstrated as well as parameterized control of changes in direction and speed. Feedback-error learning is applied to learn predictive torque models, which allows for the low-gain control that typifies many natural motions as well as producing smoother simulated motion.

## 1 Introduction

Locomotion is at the heart of many motions, both real and animated. Animated motion is most often created directly by animators using keyframing, or by capturing and then processing human motion. However, these approaches fail to scale to the very large set of possible motions that might arise in a realistic environment. For example, there are an infinite number of ways in which two characters might bump into each other or in which a character may move through a constrained, unpredictable environment. Algorithmic approaches have the potential to be more general and capable of generating families of motions rather than individual motions.

A subset of algorithmic approaches take physics into account. These include trajectory optimization methods, or, alternatively,

\*e-mail: {kk Yin|kloken|van@cs.ubc.ca}

developing controllers to drive forward dynamics simulations. Controller-based approaches have the advantage that they can synthesize motion at interactive rates and produce motion by using feedback strategies that continually adapt to the real-world as necessary.

The control of walking and other biped locomotion gaits has been of long-standing interest to the robotics and computer graphics communities. It is a challenging problem for many reasons. Walking bipeds are unstable and underactuated and their control involves high-dimensional states and high-dimensional actions. Locomotion involves joint limit constraints, torque-limit constraints, contact constraints, and contact impacts. Locomotion may have a number of contradictory goals, including robustness and energy usage. Lastly, while data-driven approaches have been very successful at generating kinematic models of locomotion, it is unclear whether such strategies can be successfully adopted to learn control strategies for dynamic simulations.

There exists a vast literature related to the control of bipedal walking, much of it in the robotics, control, and biomechanics communities. Common approaches to locomotion control include: (a) the use of passive walking as a starting point for the design of active walkers; (b) the use of “zero moment point” control; (c) using a fixed control architecture and applying parameter search to find the parameter settings that yield successful walking gaits; and (d) developing feedback laws based upon insights into balance and locomotion. Our proposed framework builds on the last of these approaches.

### 1.1 Overview of Approach

We provide a simple framework for biped control<sup>1</sup>. The starting point is the use of a simple finite state machine or *pose control graph*. Each state consists of a body pose representing target angles with respect to their parent links for all joints. All individual joints attempt to drive towards their target angles using proportional-derivative (PD) controllers. Transitions between states occur after fixed durations of time, or, for other states, after a new foot contact has been established. Walking gaits can be modeled using as few as four states, while running gaits can be modeled using as few as two states.

The pose control graph as described thus far does not have any notion of balance and thus does not produce robust locomotion. However, a small set of modifications to this basic design does result in robust locomotion. First, we require that both the torso and the

<sup>1</sup>SIMBICON is an acronym for SIMple Biped CONtrol.

swing-leg femur (i.e., swing hip) have target angles that are expressed with respect to the *world* frame, unlike the remaining links which have target joint angles expressed with respect to their parent links. In order to make the resulting torques be physically realizable without the use of external torques, the stance-hip torque is left as a free variable. Second, a feedback term is added to continuously modify the swing hip target angle as a linear function of the center of mass (COM) position and velocity. This provides a robust balancing behavior by changing the future point of support.

We can mimic the style of motion capture data by replacing the individual control states with tracking of a desired motion, using the same mix of local-and-world coordinate tracking. While such tracking normally requires high gains and a resulting stiff and reactive motion, we can apply feedback error learning (FEL) in order to produce a control solution that relies largely on predictive feed-forward torques. As a result, the final motion requires only low-gain feedback and exhibits considerably less unnatural oscillation because of the anticipatory nature of the predictive torques.

The resulting controllers can be applied to 2D and 3D bipeds of human proportion and mass distributions to produce many different styles of locomotion using physics-based forward dynamics simulations. Only physically-valid internal torques are used to produce the motion, and thus the approach may also extend to humanoid robots. Individual controllers are robust to large pushes and significant terrain variation. Controllers can be interpolated and parameterized. Direct transitions between many of the controllers are demonstrated.

## 1.2 Contributions

Our contributions are as follows. First, we integrate and build on previous insights to develop a simple new strategy for the control of balance during locomotion. We show that this can be used to develop controllers for a wide variety of 2D and 3D biped gaits. To our knowledge, we are the first to demonstrate a large set of integrated, physically-simulated bipedal skills, including many styles of walking, omni-directional push-recovery while walking, running, stylized running (scissor hop), and skipping. Second, we develop and demonstrate controller-based imitation of motion-captured gaits which exhibit robust balancing behavior. Third, we demonstrate that feedback error learning can be used to produce anticipatory, low-gain locomotion control. The simple framework opens the door to developing significantly wider sets of locomotion skills for physically-simulated characters and possibly bipedal robots.

## 2 Related Work

The wealth of literature on walking control precludes an exhaustive review. Our discussion aims to touch on the major categories of techniques, as well as focussing on the specific work that is closest to our own.

The seminal work of Raibert, Hodgins, and colleagues [Raibert 1986; Raibert and Hodgins 1991; Hodgins 1991; Hodgins et al. 1995] contains key insights into producing robust hopping and running gaits. At the heart of this research is a three-way decomposition of control of hopping height, control of torso pitch, and control of hopping speed. Swing foot placement provides the basic mechanism for controlling balance from stride to stride. The algorithms are applied to the control of running for biped robots [Raibert 1986; Raibert and Hodgins 1991; Hodgins 1991], walking for biped robots [Hodgins 1991], and running for human characters [Hodgins et al. 1995; Hodgins and Pollard 1997]. We are unaware of demonstrations of the algorithm being used to control walking for virtual

humans. We note that the control of swing foot placement is also found in other walking algorithms such as [Miura and Shimoyama 1984; Laszlo et al. 1996; Kuo 1999].

Our control framework integrates a number of the ideas from previous work, while being identical to none. We differ in several respects from the Raibert-style hopping control. The balance feedback mechanism we propose makes *continuous adaptations* during a locomotion cycle, and uses both the *position and velocity* of the center of mass. This latter information helps establish the current phase of an ongoing step, and thus it is more informative than the velocity alone. [Hodgins 1991; Raibert and Hodgins 1991] use only the velocity for control of hopping, sampled once per hop, and use a fixed step length for walking. We note that [Miura and Shimoyama 1984] is an example of an inverted pendulum control strategy that employs continuous feedback based on the inverted-pendulum body angular position and angular velocity. The framework we propose is simpler in many respects and will be demonstrated to be capable of producing a significantly larger variety of gaits.

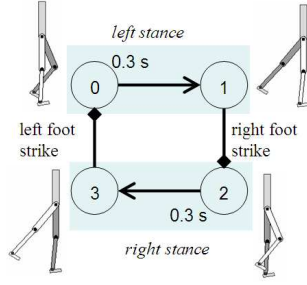
A widely-studied class of control algorithm can be developed by computing trajectories that are known to be physically feasible and therefore satisfy the zero-moment-point (ZMP) constraint [Vukobratovic and Juricic 1969]. Small disturbances to the motion can be accommodated by adjusting the ZMP dynamically during the motion. This approach has been shown to work well for the walking control of real robots [Honda Motor Co. 2006; Kim et al. 2006; Kaneko et al. 2002]. Target trajectories can be derived through an optimization process, often informed by motion capture data [Dasgupta and Nakamura 1999; Popovic and Witkin 1999]. ZMP approaches need to include swing-foot placement if they are to deal with large disturbances.

Control can also be achieved by defining a parameterized control policy and then searching for parameter values that yield appropriate control behaviors. Searching directly in the often high-dimensional parameter space is known as *policy search*, and this has been applied with some success [Taga et al. 1991; van de Panne and Fiume 1993; Auslander et al. 1995; van de Panne et al. 1994; Tedrake et al. 2004; Sharon and van de Panne 2005]. To date, these techniques have not been able to demonstrate the scalability and robustness needed to make it a useful, widely applicable control technique for animation and robotics.

Reinforcement learning (RL) offers a long-term promise of being able to learn control strategies in a principled way. It has been applied in a number of ways to the control of walking [Tedrake et al. 2004; Nakanishi et al. 2003; Morimoto et al. 2004; Smith 1998]. The high-dimensionality of the state spaces involved in the control of locomotion remains problematic, however, as does the need to design an appropriate reward function. The solutions do not exhibit the compactness and transparency which would afford control to animators in shaping the results. Many of the control strategies sample the state only once per step when computing a control decision. Work in this area is usually demonstrated on simplified models instead of humanoids, with [Smith 1998] being a notable exception.

Some strategies forego some of the physical fidelity in order to achieve desired plausible motions. One choice is to allow the addition of external forces [Wrotek et al. 2006]; another is to blend kinematic and dynamic motions [Zordan et al. 2005]. Other commercial systems use undocumented strategies [NaturalMotion 2006]. SIMBICON produces balanced locomotion behaviors using forward dynamics simulation and thus avoids the complications that mixed kinematic/dynamic solutions use.

The integration of a limited number of carefully crafted control strategies is demonstrated in [Wooten 1998; Faloutsos et al. 2001].



**Figure 2:** Finite state machine for walking

The interesting recent work of [Sok et al. 2007] parallels many of our goals and makes use of an optimization process to produce controllers for planar articulated characters that are capable of imitating motion capture data.

### 3 Balance Control Strategy

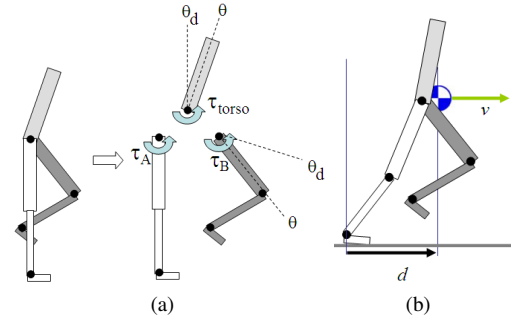
The control strategy can be described in terms of three elements: a finite state machine, torso and swing-hip control, and balance feedback. Each of these elements is now described in further detail.

#### 3.1 Finite State Machine

Our controllers are based on finite state machines, with each state having its own target pose for internal joint angles, as shown in Figure 2. For symmetric gaits, pairs of states will be left-right symmetric, e.g., states 0 and 2, 1 and 3. Transitions between states happen after an elapsed time, e.g., state transition  $0 \rightarrow 1$ , or after foot contact, e.g.,  $1 \rightarrow 2$ . If a foot contact has already occurred before entering a state having an outbound foot-contact transition, then the controller simply spends no time in that state. In any given state, the joints apply torques computed by proportional-derivative control,  $\tau = k_p(\theta_d - \theta) - k_d\dot{\theta}$ , in order to drive each joint to its desired local angle. The poses represent a desired set of joint angles and are typically not actually achieved. For example, while in state 1 in Figure 2, the biped’s pose in practice has its swing leg extended forwards. However, its target state has the swing leg extended backwards and thus has a net effect of moving the swing leg backwards and down, bringing it into contact with the ground.

#### 3.2 Torso and Swing-hip Control

The stance hip and swing hip are handled separately, as illustrated in Figure 3(a). First, there is a need to control the orientation of the torso with respect to the world frame. This can be accomplished using a virtual PD controller that operates in the world frame to compute a net torso torque  $\tau_{torso}$ , as shown in the figure. Second, there is also a need to decouple swing foot positioning from the current torso pitch angle. This is accomplished through controlling the swing hip with respect to the world coordinate frame. The swing hip torque,  $\tau_B$ , is thus also computed using a virtual PD controller that operates in the world frame. Last, there is a requirement that the virtual torques  $\tau_{torso}$  and  $\tau_B$  be realisable using only internal torques. We require that the desired value of  $\tau_{torso}$  is in fact the net torque seen by the torso,  $-\tau_A - \tau_B$ . This is accomplished by computing the stance hip torque as  $\tau_A = -\tau_{torso} - \tau_B$ .



**Figure 3:** Elements of the balance control strategy: (a) Relationship between torso, stance-hip, and swing-hip torques; (b) Center-of-mass position and velocity.

#### 3.3 Balance Feedback

The last component of the control strategy is to apply a balance feedback strategy to the swing foot placement. We employ a feedback law of the form

$$\theta_d = \theta_{d0} + c_d d + c_v v$$

to the swing hip, where  $\theta_d$  is the target angle used for PD control at any point in time,  $\theta_{d0}$  is the default fixed target angle as described in the FSM,  $d$  is the horizontal distance from the stance ankle to the center of mass (COM) as shown in Figure 3(b), and  $v$  is the velocity of the center of mass. The midpoint of the hips can be used as a simple and effective proxy for estimating the position and velocity of the center of mass. We use this simplification in both 2D and 3D.

The feedback gain parameter  $c_d$  is important for providing balance during low-speed gaits or in-place stepping. Consider a situation for an in-place (desired zero velocity) walking gait with current velocity  $v = 0$ , and two possible COM positions  $d_a = +10cm$ ,  $d_b = -10cm$ . In the first case, there is a need to step forward quickly, while in the second case there is a need to step backwards quickly in order to recover balance. The combination of  $(d, v)$  provides complete information about the current position in the gait cycle, i.e., the current phase, whereas  $v$  alone only provides information with regards to the current velocity error.

In order to extend the control scheme to 3D, the control strategy is applied in both the sagittal and coronal planes. Balance feedback in the coronal plane uses an analogous measure of  $d, v$  in order to make alterations to the lateral placement of the swing foot using the swing hip.

This form of balance feedback can be extended more generally to multiple joints using the form

$$\theta_d = \theta_{d0} + \mathbf{F} \begin{bmatrix} d \\ v \end{bmatrix} \quad (1)$$

where  $\mathbf{F}$  is an  $n \times 2$  matrix with feedback coefficients to the desired target joints. We use this more general structure to add stance ankle feedback for quiescent stance poses, for example.

In the two following sections, we describe how controllers can be manually designed and how controllers can be created using motion capture data.

## 4 Manual Controller Design

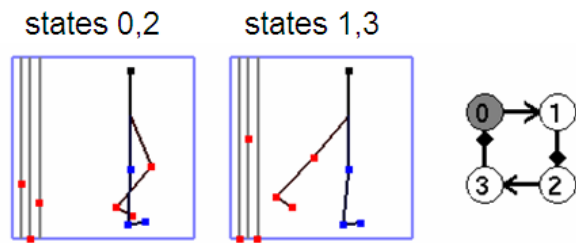
Given the controller architecture described in the previous section, we need methods for choosing the number of states and the parameters of each state. The resulting parameters should satisfy the requirements of the animator or control-system designer. Unfortunately, it is difficult to precisely pin down such requirements. Criteria for locomotion may include measures of style, robustness to perturbation, and energetic efficiency, all of which may push the solution in different directions and with design compromises that will be unknown in advance. Therefore, before resorting to more complex schemes, we first investigate manual interactive design of the required finite state machine.

The control parameters can be grouped into several categories: (a) number of states and state-transition parameters; (b) the balance feedback gains,  $c_d$  and  $c_v$ ; (c) the target poses for each state; (d) the initial state for using the controller; and (e) the joint limits, torque limits, and PD-controller gains. In our work, we fix the parameters belonging to category (e) and document these in the results section. The remainder of our discussion focuses on the other parameters.

The choice of the number of states reflects the detail with which to model the various phases of a locomotion gait. We use four states to model our walking gaits, consisting of two symmetric walking steps. Each step has two states, the first of which lifts the swing foot upwards and forwards for a fixed duration of time, and the second of which drives the swing foot towards the ground until contact is made. This model is capable of many different walking styles, both forwards and backwards. The FSM states also serve as a coarse model of the phase of the motion when switching between controllers. Thus, if a request is made to switch from one walk style to another, this is done by transitioning from state  $n$  of one controller to state  $n + 1$  of another controller. For this reason, while our running gaits can be modeled using simple two-state controllers (one for each running step), we add two zero-duration dummy states in order to have the same four-state structure as for the walking gaits. This allows for transitions between walking and running gaits. Our skipping controller will have 8 states, reflecting its more complex sequence of actions.

We begin controller designs using the planar biped model, and then use the resulting parameters as a starting point for the design of corresponding 3D controllers. We use a graphical user interface (GUI) to allow a user to directly explore the parameters settings associated with each of the controllers states, as shown in Figure 4. Users can immediately observe the effect of parameter changes reflected in an ongoing simulation. Three sliders on the left of each state GUI are used to set the state duration,  $c_d$ , and  $c_v$  parameters. The target pose parameters are set by using the handle points on the stick figure. The target poses for the torso and the swing femur are interpreted with respect to the world frame. The target pose angle for the stance femur is ignored by the controller, given that the stance hip torque is treated as a free parameter whose value is determined from the torso and swing-hip torques. All the remaining joint angles define target angles with respect to their parent’s coordinate frame. The interface readily exposes the key-frame like nature of many of the controller parameters.

The most important parameters for each state are the state duration  $\Delta t$  and the target angles for the swing hip and swing knee. Because the resulting motion style is most heavily dependent on only these three parameters per state, it becomes relatively easy for users to interactively explore their settings to yield desired motions. The ankles make a significant contribution to some styles, such as the skipping gait. The stance knee is usually almost straight. The torso is usually desired to be vertical. The balance feedback gains are set in a similar fashion across many of our controllers.



**Figure 4:** Graphical interface for adjusting controller parameters. Sliders on the left control  $\Delta t$ ,  $c_d$ , and  $c_v$ .

The design of a stepping-in-place gait for 2D locomotion represents a good starting point that can then be modified for the design of other motions. The target angles, as shown in Table 1, look very much like a simple pair of keyframes, one in a standing posture, and another with the swing leg in the air in a bent pose as one might expect for stepping-in-place. This leaves very few remaining parameters to set, principally the duration of the leg lift pose and the balance feedback gains for the swing hip.

Small changes ( $\pm 15^\circ$ ) to the desired torso pitch can be easily accommodated by treating the extra torque produced by gravity as a disturbance. During locomotion, the torso may exhibit a somewhat unnatural bobbing motion. This is the result of the torso servo always reacting to the motion of the hip, rather than anticipating it. We address this in the section on feedback-error learning.

A reasonable choice of initial state is required in order for a controller to function as designed. In practice, the balance feedback terms endow the controllers with relatively large basins of attraction, as demonstrated by their robustness to external pushes and changes in terrain, and the ability to transition directly between many of the controllers. We begin our walking controllers from a double stance state with a moderate forward velocity ( $1m/s$ ), although we note that our basic forward walking controller can begin just as well from rest. We note that symmetric controllers can exhibit asymmetric gaits from some initial states while producing symmetric gaits from other initial states. This difficulty can be overcome by using initial states closer to the desired limit cycle.

## 5 Controllers from Motion Capture Data

An alternative to manual design is to use motion capture data as the basis for developing a controller. This allows a kinematic motion to be imported into a dynamic setting. Whereas kinematic motion capture data cannot be made to stumble for an unseen step or respond to a push, a style-mimicking controller allows for these effects.

We develop style-mimicking controllers as follows. Given 3–7 cycles of motion capture walking data, we apply Fourier analysis to a representative joint, such as the right hip, in order to extract the period  $T$  of the walking cycle, which is estimated using the primary frequency  $\omega$  of the Fourier transformation. We then select the next several largest Fourier coefficients corresponding to frequencies that are an integer multiple of  $\omega$ . When the reconstruction error reaches a preselected accuracy, we discard the remaining Fourier components. This filters the original motion capture data to a smooth periodic motion  $\Theta$  that reflects an averaged walk cycle of the given style and represents it using a compact set of coefficients.

The extremal points of  $\Theta$  are detected automatically. The time  $t_m$  of the largest right hip flex is assigned a fixed phase  $\phi(t_m) = 0.25$  in the walk cycle, where  $\phi \in [0, 1]$ . States 1 and 3 are discarded from the FSM in Figure 2. The remaining two states serve the purpose of

differentiating between the left-stance and the right-stance phases. The transition between the states occurs on foot contact, which is expected to occur at approximately  $\phi = 0.5$  and  $\phi = 1$ . The double-stance phase is considered to be part of the stance phase that has just begun.

The trajectory  $\Theta$  serves as a target trajectory in place of the target poses used in the manually-designed controllers. Within each state, the joint angles individually track the motion capture trajectories  $\theta_{d0} = \Theta(\phi(t, T))$  using PD-controllers. The phase is reset to 0 or 0.5 upon transitioning to the next state. As is the case for the manually-designed controllers, the torso angle and swing-hip angle do their tracking with respect to the world frame. Similarly, the stance hip does not track its motion-captured counterpart and, as before, its torque is computed from the known torso and swing-hip torques. The PD-controller, however, is changed from  $\tau = k_p(\theta_d - \theta) - k_d\dot{\theta}$  to  $\tau = k_p(\theta_d - \theta) - k_d(\dot{\theta} - \dot{\theta}_d)$ . The  $\dot{\theta}_d$  term helps in tracking the target trajectory with minimal time lag.

Controllers based on motion capture data apply balance feedback to both the swing hip and, for slow walks, to the stance ankle using Equation 1. While a fairly broad range in values result in stable gaits, we currently tune these by hand in order to yield a robust gait and a strongly-attracting limit cycle which will be required for the successful application of feedback error learning, as will be discussed later.

The controller will not perfectly imitate the motion capture reference motion for a number of reasons. First, the original motion capture data may contain noise from data capture and data processing and may not be dynamically consistent. Second, the physical system parameters of the simulated human may not exactly match that of the motion-capture actor. This includes link dimensions, joint placement, mass and inertial parameters, and joint gains. Additionally, we forego tracking of the stance hip angles in order to insert the balance-feedback mechanism. The resulting motion is thus encouraged to imitate the overall style of the reference motion, but does not precisely match other parameters that could also be used to characterize gaits, such as step-length or velocity. Lastly, in order to closely follow the reference motions, the tracking control requires high-gain PD controllers. These can be lowered using the feedback-error learning scheme discussed in the following section.

## 6 Feedback Error Learning

The controllers described thus far produce motions that are quite robust to disturbances and are able to closely track reference trajectories. However, the high gains used in the feedback loops are representative of stiff movements. Also, the torso pitch angle oscillates about its desired position because it is always reacting to the movement of the hip rather than anticipating it.

Human motor control commonly increases the mechanical impedance of the control as a strategy for producing robust motions in environments where perturbations are expected, and uses low-impedance control when the environment is highly predictable[Takahashi et al. 2001]. The low-impedance control in predictable environments can be achieved by using anticipatory “feed forward” torques together with low-gain feedback. The latter is still necessary to deal with small deviations from the desired trajectory that may always be expected. The feed-forward torques are also thought to be necessary in order to deal with the slowness of the human nervous system, as compared to the fast response that would be required for purely feedback-based control.

Previous work using tracking of upper-body motion capture trajectories[Zordan and Hodgins 2002] requires high-gains in order to track well. At the instant of an unexpected impact, the gain is

lowered for a short duration in order to mimic the low-impedance control normally exhibited during skilled motion, before being increased again to resume tracking. The ephemeral low-gain portion of the motion is implausible in that it would not track the default motion in the absence of the disturbance. In order to address this, [Yin et al. 2003] use inverse dynamics in order to estimate the feed-forward torques that would normally be in effect during skilled motion, although they do not deal with issues of balance. In contrast, we apply feedback error learning (FEL) in order to learn feed-forward torques, which then allow our controllers to operate with low tracking gains. Because the low-gain motions are less robust than the high-gain motions, we can optionally *increase the gains* for some time after impact after a reaction-time delay (150ms) in order to simulate a natural perturbation-recovery reaction.

Feedback error learning is a form of adaptive control[Kawato et al. 1987; Nakanishi and Schaal 2004] and allows for the learning of the inverse dynamics of a system in order to reproduce given motion trajectories. In its most general form, the feed-forward function learns  $\tau = f(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}})$ , where  $\mathbf{x}, \dot{\mathbf{x}}$  is the current system state (positions and velocities) and  $\ddot{\mathbf{x}}$  is the vector of commanded accelerations. We move away from this general form and instead learn the feed-forward torques as a function of the current phase of the motion:  $\tau = f(\phi)$ . We estimate  $\phi$  using  $\phi = t/\hat{T}$ , where  $t$  is the current elapsed time in a given state and  $\hat{T}$  is the current estimate of the period. For repetitive motions such as a walk cycle, these simplifications work well. To our knowledge, FEL has not been successfully applied to a dynamical system as complex as a fully simulated virtual character capable of a variety of robust locomotion behaviors.

Our implementation of FEL divides the phase  $\phi$  uniformly into  $N$  bins. We use  $N = 20 - 1000$ . The current phase bin is given by  $n = tN/\hat{T}$ . Each bin uses a low-pass filter of the form

$$v'_{ff} = (1 - \alpha)v_{ff} + \alpha(v_{ff} + v_{fb}) \quad (2)$$

to update the feed-forward torques. Here  $v_{ff}$  and  $v_{fb}$  represents the feed-forward and feed-back torques applied for that phase of the motion corresponding to bin  $n$ . We use a learning rate of  $\alpha = 0.1$ . The feed-forward values are initialized to zero. During the learning process (and afterwards), it is the sum of feed-forward and feedback torques that is applied, i.e.,  $v_{ff} + v_{fb}$ . FEL can be applied to one joint at a time, or to all joints simultaneously.

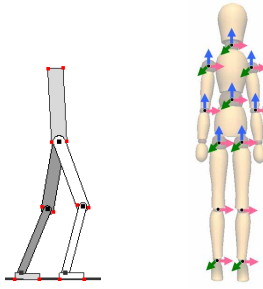
High learning rates may yield to convergence problems because the use of feed-forward torques will influence the resulting motions. While it is difficult to obtain analytic guarantees of convergence of FEL for complex dynamical systems, we have not experienced any convergence problems with our chosen learning rate. It is also useful to limit the magnitude of the feed-forward torques because some predictable disturbances can never be fully accommodated. An example of this is the force impulse caused by foot contact, which instantaneously creates a small change in angular velocity for the torso and requires an equivalently instantaneous control impulse in order to fully ensure that the torso never exhibits any pitch.

## 7 Results and Discussion

We apply the SIMBICON framework to simulated 2D and 3D bipeds having human-like proportions and mass distributions. Figure 5 shows the models and their degrees of freedom.

The 7-link planar biped has a 70 kg trunk, 5 kg upper legs, 4 kg lower legs, and 1 kg feet. The respective largest dimensions are 48 cm, 45 cm, 45 cm, and 20 cm. A combined head-arms-trunk (HAT) model is used, as is common in the walking simulation literature. The 2D biped is simulated using an optimized version of the





**Figure 5:** Degrees of Freedom (DOF) of models. Left: 2D model with 6 internal DOFs, 9 DOF in total. Right: 3D model with 28 internal DOFs, 34 DOF in total.

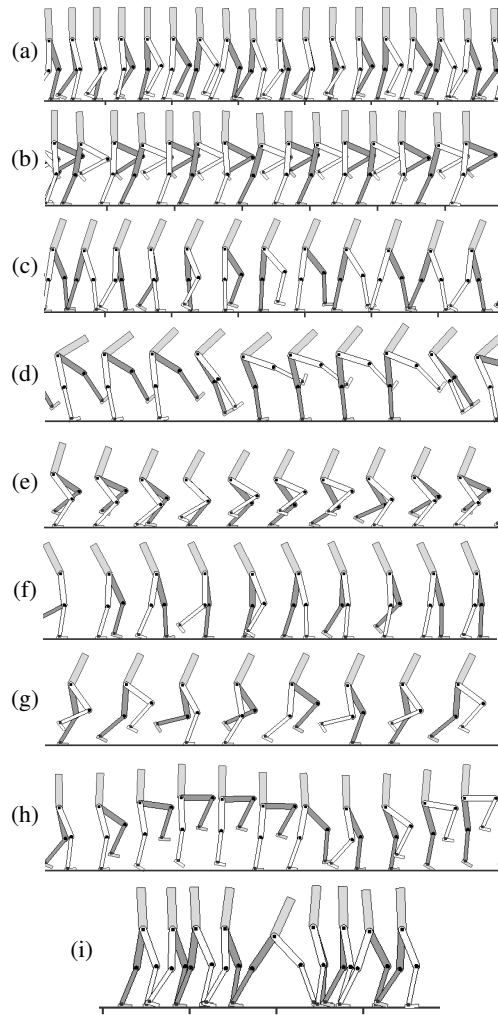
Newton-Euler equations of motion. A spring-and-damper penalty-force ground contact model is applied to points at the front and back of the feet. PD gain values of  $k_p = 300 \text{ Nm/rad}$ ,  $k_d = 30 \text{ Nms/rad}$  are used for all joints. Joint limits are enforced on the hips and knees. Ground stiffness parameters are  $k_p = 100000 \text{ N/m}$ ,  $k_d = 6000 \text{ Ns/m}$ . We use a Coulomb friction model with a friction coefficient of 0.65. A time step of  $0.0001 \text{ s}$  is used. We use torque limits of  $1000 \text{ Nm}$ , which can be lowered to  $370 \text{ Nm}$  for all motions except the fast run. The basic walk controller supports a torque limit of  $90 \text{ Nm}$ , below which it becomes weak-kneed and falls. With control, the simulation runs 5 times faster than real-time unoptimized on a 1.8 Ghz CoreDuo PC.

The parameters for the 3D biped model are the same as used in [Laszlo et al. 1996]. We scale the limb lengths to match our motion capture subject. The 3D biped is simulated using Open Dynamics Engine[ODE ] version 0.6. Our simulation time step is  $0.005 \text{ s}$ . Contacts are modeled using constraints and an approximated Coulomb friction cone, solved as a linear complementarity problem (LCP). We use a friction coefficient of 0.8, which is typical for a rubber sole. The coefficient of restitution for collisions is assumed to be zero. Torque magnitudes are limited to  $k_p$ . The largest  $k_p$  value is 1000, used for the waist joints. All other joints use  $k_p = 300$  or less. We use  $k_d = 0.1k_p$ . With control, the unoptimized simulation runs 1.2 times faster than real time on a 3.2 Ghz Pentium 4.

## 7.1 2D Biped Locomotion

A set of 12 periodic gaits have been authored using the GUI (§ 4). The parameters for these controllers are given in Table 1. We designed these gaits to achieve a wide variety of motion styles using a small number of states. They have not been designed to be optimal gaits with respect to any given criterion. We have also authored acyclic controllers for stopping and remaining balanced on two feet, stopping and remaining balanced on one foot, and taking a single large step in the middle of a longer walking sequence. A subset of the motions are illustrated in Figure 6.

The running controllers do not have a strong preference to run at a particular speed. As such, they can be ‘pushed’ to run at various speeds. Parameterized control of speed is likely feasible, although was not investigated. As compared to [Raibert and Hodgins 1991], our framework uses no explicitly-computed injection of energy to maintain a given flight time during running or skipping. The control laws are identical for all FSM states, modulo the change of legs fulfilling stance-leg and swing-leg roles, and are governed by the target angles and feedback gains. This supports a style of running that, qualitatively speaking, looks less like hopping than previous work[Hodgins et al. 1995; Hodgins and Pollard 1997].

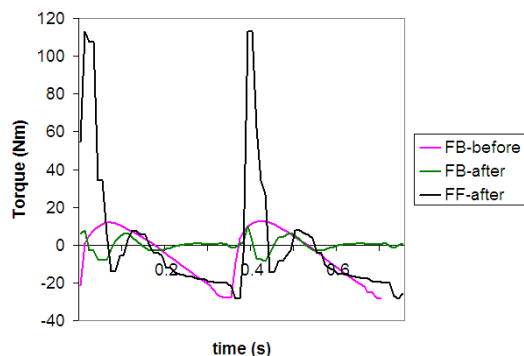


**Figure 6:** A subset of the manually-designed controllers for the planar biped. (a) walk; (b) high-step walk; (c) bent walk; (d) scissor hop; (e) crouch walk; (f) backwards walk, right-to-left; (g) fast run; (h) skipping; (i) big step.

## Controller Transitions

Controllers are bound to keystrokes and it is possible to interactively request transitions between the controllers. This is accomplished during state transitions, jumping from state  $n$  of controller A to state  $n + 1$  of controller B. An example of the kind of transitions that be successfully executed is as follows: *walk* → *in-place walk* → *stand* → *back-flip* → *in-place walk* → *walk* → *big-step* → *walk* → *in-place walk* → *high-step walk* → *in-place walk* → *scissor hop* → *walk* → *skip* → *walk* → *bent walk* → *walk* → *crouch walk* → *walk* → *in-place walk* → *backwards walk* → *in-place walk* → *back-and-forth stepping* → *walk* → *run* → *fast run*. A portion of this is shown in the video that accompanies this paper.

In the absence of specially-designed transition motions, not all controller transitions are feasible. For two controllers whose motions are significantly different, the basins of attraction for each may not overlap as needed for direct transitions. Direct transitions are possible between many of the walking and running gaits. However, some gaits are significantly more sensitive to the required initial state, such as the backwards walk which first requires transitioning to the in-place walk. The “scissor hop” is particularly sensitive to its



**Figure 7:** Feedback error learning (FEL) applied to the 2D biped torso. Illustrated are the feedback torques, before and after FEL, and the learned feed-forward torque.

initial state. Its basin of attraction does not fully overlap the limit cycle of the in-place walk, and thus requires being in a particular phase of its walk cycle in order to ensure a successful entry transition. The controllers for standing on one foot also has a very limited basin of attraction due to its need to balance without stepping.

### Robustness

The robustness of the walk controller to variations in terrain is shown in Figure 1(a). The terrain includes unanticipated downward steps of 20cm and slopes of  $\pm 6$  degrees. The robustness of the gaits with respect to unanticipated pushes was tested by applying 10 pushes at 5s intervals, which serves to sample various phases of the gait while also allowing the biped time to fully recover between pushes. Any single stumble from which the biped cannot recover is deemed a failure. The walking controller can withstand 0.1s duration pushes of up to 600N forwards and 500N backwards at all 10 sampled points in the locomotion cycle. Other gaits are more sensitive to disturbances. For example, the skipping gait can withstand 0.1s duration pushes of up to 40N forwards and 50N backwards. Larger pushes, as measured by their induced change in momentum,  $F\Delta t$ , can be sustained by increasing  $\Delta t$  and decreasing  $F$ . Specific portions of the walk cycle can also withstand larger pushes.

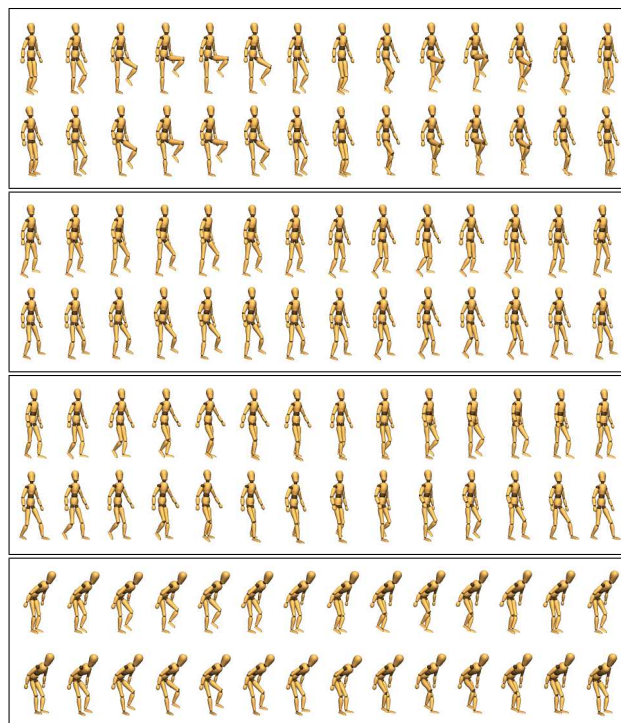
### Feedback Error Learning

The application of feedback error learning to the 2D biped torso decreases its oscillation amplitude from  $5^\circ$  to  $0.5^\circ$  for a walking gait and unchanged feedback gains. Figure 7 illustrates the learned torque and the decreased amplitude of the feedback torques. Foot-strike events occur at time  $t = 0$  and  $t = 0.36$  on the graph and cannot be fully anticipated due to their impulsive nature.

## 7.2 3D Biped Locomotion

### Manual Controller Design

Controllers for 3D walking require twice the number of parameters as for 2D control because of the need for lateral (coronal-plane) control. Manually-designed controllers have been developed for two-foot hopping, three styles of forward walking, walking up a slope of 20 degrees, two styles of forward running, a sideways-walk Tai Chi movement, and skipping. Direct transitions are possible between most of the walking and running gaits. Many of our 2D control strategies work directly when applied to the 3D model and applying a common set of feedback gains for the lateral hip movement that is responsible for lateral balance. There is some variation in the style of the 2D and 3D motions, likely because our 2D and 3D models have different masses and proportions. For the few cases



**Figure 8:** Imitation of motion capture data. Each box compares the original motion (top) and the controller motion (bottom). The boxes from top to bottom: high knee walk; wide stance walk; backwards walk, right-to-left; bent-forwards walk.

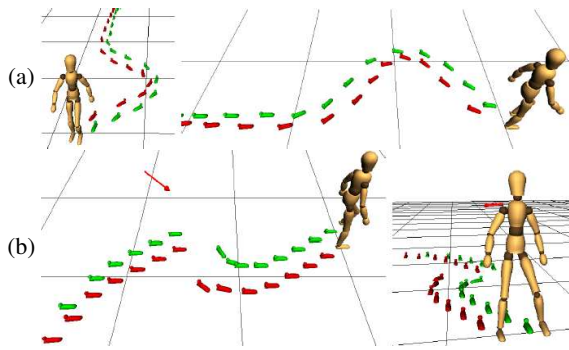
where the difference in models is problematic, minor adjustments to target angles and gains are sufficient to achieve a functional 3D motion.

### Motion Capture Imitation

We have developed seven controllers from motion capture data, including four different types of in-place walking (normal, wide-stance, bent-trunk, high-knee), forwards walking, backwards walking, and sideways walking. Figure 8 shows comparisons of the original captured motions and the motion resulting from the controllers.

If necessary, we use a small amount of manual tuning to make the tracking-based controller functional. Some of the original in-place stepping gaits exhibit only a small amount of swing-foot clearance during the step. This can cause a failure in the tracking-based controller to lift the swing foot off the ground. A simple correction is to add a constant swing-hip offset so that the swing foot lifts off the ground as desired. Manual tuning of the balance feedback gains for the swing hip, both sagittal and lateral, is sometimes necessary. Section 7.3 describes the types of artifacts that are seen when the gains are misadjusted. The in-place high-stepping walk requires use of sagittal balance feedback gains for the stance ankle because of the time spent balanced on the stance foot and the large shift in the COM caused by the high lift of the swing leg. An unnatural aspect of some of our walking results is that they may fail to properly mimic aspects of the ankle motion and foot toe-off. We speculate that this may be resolved with additional tuning of the ankle PD-gains and ankle balance feedback gains. We have not yet tried to replicate running motions from motion capture data, although we are optimistic that this would work.

We speculate that there will be several categories of motions where our motion-capture-to-controller methodology will fail. Acrobatic



**Figure 9:** Variations in locomotion, illustrated using footprints. (a) Turning behavior applied to a slow forward walk controller reconstructed from motion capture data. (b) A wide-stance in-place walk reconstructed from motion capture data is made to walk diagonally forward and to the left through the simple addition of offsets to the swing-hip target angles. It then reacts to a large push diagonally to the right.

motions have significant flight phases and therefore rely on accurately achieving specific linear and angular momentum upon take-off. Our balance control feedback does not provide these. Dynamic motions that do not involve periodic stepping motions are likely to be problematic. Lastly, dynamic motions that do not involve any stepping require a ZMP approach or an approach that can exploit other parts of the body to help maintain balance.

### Parameterization

The robust nature of our controllers means that once a controller has been constructed, either manually or from motion capture data, additional control strategies can then be layered on top. High-level controllers can be developed to control walking styles or walking directions. Much of this kind of control can be added in an intuitive fashion as displacements to the target poses or target motions. For example, to add a lateral component to straight line walking, we add an antisymmetric displacement angle to the lateral target hip angles for all the poses in the PCG. If instead we add symmetric displacement angles to each lateral hip angle, this produces a straight walk with altered stance width. The result is shown in Figure 9(b), with the addition of an external push.

Controllers can cope with unanticipated gentle slopes and small steps. Steeper slopes or larger steps require adding a displacement  $\Delta\theta_{hip} = k\theta_s$  to the target hip angles to insure foot clearance, where  $\theta_s$  is the angle of the slope. For sufficiently steep slopes, the ankle angles also need to be adapted.

Turning is generated by modulating the desired facing angle. The desired facing direction in Figure 9(a) is varied according to  $0.9 * \sin(\pi t/12)$  in radians. The stance hip is then used to achieve the desired facing direction. A light backpack can be worn with an unchanged, vertical torso target pitch. Heavier backpacks need to be accommodated by pitching the torso forward accordingly. Interpolation between gaits can be achieved by interpolating between parameters of the corresponding controller states. Since the balance controller only uses the lower limbs, the upper body is left free for different styles or additional tasks. For example, we can choose to keep the arms straight down or to swing them naturally.

### Robustness

A robust balance controller also means that the locomotion can deal with unexpected environmental disturbances automatically. The largest recoverable pushes as measured in eight evenly-sampled di-

rections are (0, 340), (230, 230), (330, 0), (220, -220), (0, -270), (-190, -190), (-240, 0), (-190, 190), where each pair defines the (lateral, sagittal) push magnitudes in Newtons. The pushes are applied at chest height at a phase angle of  $\phi = 0.1$  and have a duration of 0.4s. These numbers are comparable with specially developed push recovery controllers [Kudoh et al. 2006], which are only demonstrated in the sagittal plane and are computed offline using quadratic programming. We also tested the robustness with respect to kinematic and dynamic model variations. For example, we have increased the femur length by 10% for the walking gait defined in Table 1 while maintaining both the style and stability of the gait. Larger changes can be accommodated, first resulting in a change of style while still being robust. Some gaits are particularly sensitive to some parameters. For example, the fast running gaits tend to be sensitive to the balance feedback gains. Stairs can cause problems because the controllers cannot “see” an upcoming step, and the resulting toe stub or ill-placed foot can cause a fall.

### Feedback Error Learning

Feedback error learning has been successfully applied to the upper-body joints and the virtual torso torque for all manually-designed FSM controllers. We limit the maximum feedforward torques to be  $|\Delta\theta k_p|$ , where  $k_p$  is the PD spring constant and  $\Delta\theta = 0.2$  radians. The feedforward torque will thus be capable of eliminating oscillations of approximately 0.2 radians in magnitude. We experiment with various resolutions for representing the feedforward torque, using phase bins that correspond to  $\Delta t_\phi \in [0.5, 25]$  ms. For the 3D walk controller given in Table 1, we test several different resolutions and computed the energy ratio of the feedback torques to the feed-forward torques,  $r = \frac{\int \|fb\|}{\int \|ff\|}$  after FEL. For  $\Delta t_\phi = 0.5$  ms,  $r \approx 2.5\%$ ;  $\Delta t_\phi = 5$  ms,  $r \approx 3.5\%$ ;  $\Delta t_\phi = 25$  ms,  $r \approx 14\%$ . Because  $\theta_{d0}$  is discontinuous for the lower-body joints for the manually-designed controllers, we do not apply FEL to these joints. The large discontinuities in the desired joint angles that occurs upon transitioning to a new FSM state are not suitable for modeling directly using a feed-forward torque. This can be circumvented by treating the output of a simulation as being the equivalent of motion capture data, and applying the strategy that we shall describe next.

Feedback error learning can also be applied to controllers that track motion capture data. FEL can be applied directly to the upper-body joints and the virtual torso torque. Applying FEL to the lower limbs requires two adaptations to the basic FEL learning process. First, for all joints using balance feedback, the joint angle tracking torques need to be decoupled from the balance-feedback torques. Therefore it is essential to apply FEL based only on the component of the torque that is used to track the joint angle target trajectory, and *not* the component that is added to achieve balance control. Second, despite the use of smooth target-angle trajectories, there remains a discontinuity at the instant of stance-leg/swing-leg exchange. To accommodate this, we adapt the desired trajectory towards the realized simulation trajectory using a displacement trajectory. The displacement trajectory  $\delta\theta$  is initialized to zero and is modified over time according to  $\delta\theta' = (1 - \alpha)\delta\theta + \alpha(\theta - \theta_d)$ . Each of  $\delta\theta, \theta, \theta_d$  are functions of phase and are modeled using phase bins in the same way as feed-forward torques. It can be seen that  $\delta\theta$  remains unchanged when it correctly predicts the tracking error  $\theta - \theta_d$ .

In the accompanying video, we show several comparisons between walks based on motion capture data, simulated using (a) feedback control alone, and (b) using combined feedback and feedforward control. The latter motions exhibit smoother, more stable motion while having PD constants  $k_p, k_d$  that are the same or lower.



### 7.3 Setting the Balance Feedback Gain Parameters

$c_d, c_v$  are usually within the range of  $[0, 1]$ . For our basic 3D walk controller we use  $c_d = 0.5$  and  $c_v = 0.2$  for the swing hip in all states, in both the coronal and sagittal planes. We provide a brief summary of our bifurcation stability analysis because it provides an indication of the sensitivity of the results with respect to some of the parameters, and we also note that implementors may observe phenomena such as period doubling [Vakakis and Burdick 1990; Koditschek and Buhler 1991]. Beginning from a fixed initial state, we change one selected parameter across all four states to find its viable range, while fixing all other parameters to their nominal values. The stable range of the parameters are:  $[-0.71, 1.4]$  for  $c_d$  and  $[0.03, 0.59]$  for  $c_v$  in the sagittal plane;  $[-1.29, 1.13]$  for  $c_d$  and  $[-0.06, 0.48]$  for  $c_v$  in the coronal plane. For most operating points within these ranges, stable limit cycles can be achieved. For operating points near the upper limits, period doubling and chaotic behavior develop on occasion. When  $c_v$  is below the lower limit, the velocity of the character accumulates until it falls. When  $c_v$  is above the upper limit, usually the character will rock back and forth (or left and right) with increasing amplitudes until the oscillations destroy the walking.

### 7.4 Limitations

The pipeline for producing controllers from motion capture data is not fully automated in that we still manually tune the required feedback gain constants, and has only been tested on styles of walking. The current gaits are not optimized for energy efficiency. We do not model the reaction-time delays of human motion. As a result, some of our motions are stable in a way that human motions may not be.

The available suite of mathematical tools for the stability analysis of high-dimensional, non-linear dynamical systems is limited. Two practical options for analysis are to work with a simplified version of the system dynamics, or to rely on simulation-based experiments. We have chosen the latter option.

## 8 Conclusions

The control of bipedal locomotion is a challenging problem. The need in animation to model multiple gaits, stylized motions, reaction to variable terrain, and reactions to external forces exacerbate this challenge. The framework presented in this paper addresses many of these challenges. We have further shown how to develop a variety of walking controllers from motion capture data and how to implement feedback error learning to achieve motions that are driven by feed-forward torques and low-gain feedback.

There are a large number of directions that can be pursued. We wish to develop libraries of ‘downloadable skills’ that can be shared. This requires file formats for exchanging controllers which describe both the controller itself and its basin of attraction [Faloutsos et al. 2001]. We wish to apply the control schemes to humanoid robots. Basic locomotion skills should be integrated with other skills that let the simulated characters interact with their environment in a rich variety of ways. Methods are needed for planning motions using the controllers we have developed.

### Acknowledgements

This research was supported by NSERC. The authors gratefully acknowledge the inspired interest of Dana Sharon and Joe Laszlo in the control of walking and without whom this work would not have been brought to fruition. We also thank Peng Zhao for discussions and help with the 3D simulation software, and the anonymous reviewers for their many helpful comments.

## References

- AUSLANDER, J., FUKUNAGA, A., PARTOVI, H., CHRISTENSEN, J., HSU, L., REISS, P., SHUMAN, A., MARKS, J., AND NGO, J. T. 1995. Further experience with controller-based automatic motion synthesis for articulated figures. *ACM Trans. on Graphics 14*, 4, 311–336.
- DASGUPTA, A., AND NAKAMURA, Y. 1999. Making feasible walking motion of humanoid robots from human motion capture data. In *Robotics and Automation*, vol. 2, 1044–1049.
- FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. *Proc. ACM SIGGRAPH*, 251–260.
- HODGINS, J. K., AND POLLARD, N. S. 1997. Adapting simulated behaviors for new characters. In *Proceedings of SIGGRAPH '97*, 153–162.
- HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 71–78.
- HODGINS, J. K. 1991. Biped gait transitions. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- HONDA MOTOR CO., L., 2006. Studies of leg/foot functions of the robot, <http://world.honda.com/asimo/p3/technology/>.
- KANEKO, K., KANEHIRO, F., KAJITA, S., YOKOYAMA, K., AKACHI, K., KAWASAKI, T., OTA, S., AND ISOZUMI, T. 2002. Design of prototype humanoid robotics platform for HRP. *IEEE/RSJ Intl. Conf. on Intell. Robots and Systems*.
- KAWATO, M., FURUKAWA, K., AND SUZUKI, R. 1987. A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics 57*, 3, 169–185.
- KIM, J., PARK, I., AND OH, J. 2006. Experimental realization of dynamic walking of the biped humanoid robot KHR-2 using zero moment point feedback and inertial measurement. *Advanced Robotics 20*, 6, 707–736.
- KODITSCHKE, D., AND BUHLER, M. 1991. Analysis of a Simplified Hopping Robot. *Intl J. of Robotics Research 10*, 6, 587.
- KUDOH, S., KOMURA, T., AND IKEUCHI, K. 2006. Stepping motion for a humanlike character to maintain balance against large perturbations. In *Proc. of Intl Conf. on Robotics and Automation*, 2661–2666.
- KUO, A. 1999. Stabilization of Lateral Motion in Passive Dynamic Walking. *Intl J. of Robotics Research 18*, 9, 917.
- 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 *Proceedings of ACM SIGGRAPH*, 155–162.
- MIURA, H., AND SHIMOYAMA, I. 1984. Dynamic Walk of a Biped. *Intl J. of Robotics Research 3*, 2, 60–74.
- MORIMOTO, J., CHENG, G., ATKESON, C. G., AND ZEGLIN, G. 2004. A simple reinforcement learning algorithm for biped walking. In *Proc. IEEE Int'l Conf. on Robotics and Automation*.
- NAKANISHI, J., AND SCHAAL, S. 2004. Feedback error learning and nonlinear adaptive control. *Neural Networks 17*, 1453–1465.
- NAKANISHI, J., MORIMOTO, J., ENDO, G., CHENG, G., SCHAAL, S., AND KAWATO, M. 2003. Learning from

demonstration and adaptation of biped locomotion with dynamical movement primitives. In *Workshop on Robot Learning by Demonstration, IEEE Int'l Conf. Intelligent Robots and Systems*.

NATURALMOTION, 2006. <http://www.naturalmotion.com>.

ODE. Open dynamics engine. <http://www.ode.org>.

POPOVIC, Z., AND WITKIN, A. 1999. Physically based motion transformation. In *Proceedings of ACM SIGGRAPH*, 11–20.

RAIBERT, M. H., AND HODGINS, J. K. 1991. Animation of dynamic legged locomotion. In *Proc. SIGGRAPH '91*, 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 *International Conference on Robotics and Automation*.

SMITH, R. 1998. *Intelligent Motion Control with an Artificial Cerebellum*. PhD thesis, University of Auckland.

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

TAGA, G., YAMAGUCHI, Y., AND SHIMIZU, H. 1991. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics* 65, 147–159.

TAKAHASHI, C. D., SCHEIDT, R. A., AND REINKENSMEYER, D. J. 2001. Impedance Control and Internal Model Formation When Reaching in a Randomly Varying Dynamical Environment. *J. Neurophysiology* 86 (Aug).

TEDRAKE, R., ZHANG, T. W., AND SEUNG, H. S. 2004. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *IEEE Intl Conf. on Intelligent Robots and Systems*.

VAKAKIS, A., AND BURDICK, J. 1990. Chaotic motions in the dynamics of a hopping robot. *Proc. IEEE Intl Conf on Robotics and Automation*, 1464–1469.

VAN DE PANNE, M., AND FIUME, E. 1993. Sensor-actuator networks. In *Proceedings of ACM SIGGRAPH*, 335–342.

VAN DE PANNE, M., KIM, R., AND FIUME, E. 1994. Virtual wind-up toys for animation. In *Graphics Interface*, 208–215.

VUKOBRATOVIC, M., AND JURICIC, D. 1969. Contribution to the synthesis of biped gait. In *IEEE Transactions on Biomedical Engineering*, vol. 16. 1–6.

WOOTEN, W. L. 1998. *Simulation of leaping, tumbling, landing, and balancing humans*. PhD thesis, Georgia Institute of Technology.

WROTEK, P., JENKINS, O. C., AND MCGUIRE, M. 2006. Dynamo: dynamic, data-driven character control with adjustable balance. In *sandbox '06: Proc. of the 2006 ACM SIGGRAPH Symposium on Videogames*, 61–70.

YIN, K., CLINE, M. B., AND PAI, D. K. 2003. Motion perturbation based on simple neuromotor control models. In *Proceedings of Pacific Graphics*.

ZORDAN, V. B., AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *Proc. ACM SIGGRAPH/EG Symp. on Computer animation*, 89–96.

ZORDAN, V., MAJKOWSKA, A., CHIU, B., AND FAST, M. 2005. Dynamic response for motion capture animation. *Proc. ACM SIGGRAPH 2005* 24, 3, 697–701.

state	$\Delta t$	$c_d$	$c_v$	tor	swh	swk	swa	stk	sta
walk									
0,2	0.30	0.00	0.20	0.0	0.40	-1.10	0.20	-0.05	0.20
1,3	fc	2.20	0.00	0.0	-0.70	-0.05	0.20	-0.10	0.20
in-place walk									
0,2	0.30	0.00	0.40	0.0	0.62	-1.10	0.20	-0.05	0.20
1,3	fc	1.55	0.00	0.0	-0.10	-0.05	0.20	-0.10	0.20
fast walk									
0,2	0.27	0.00	0.20	-0.1	0.73	-1.83	0.20	-0.05	0.20
1,3	fc	2.00	0.00	-0.1	-0.70	-0.05	0.20	-0.10	-0.06
highestep walk									
0,2	0.30	0.00	0.20	0.0	1.00	-2.40	0.20	-0.05	0.20
1,3	fc	2.00	0.00	0.0	-0.70	-0.05	0.20	-0.10	0.20
half bent walk									
0,2	0.23	0.00	0.20	-0.2	0.62	-1.10	0.00	-0.05	0.00
1,3	fc	0.60	0.00	-0.2	-0.10	-0.05	0.00	-0.10	0.00
bent walk									
0,2	0.30	0.00	0.20	-0.6	0.80	-1.10	0.00	-0.05	0.00
1,3	fc	0.60	0.00	-0.6	-0.10	-0.05	0.00	-0.10	0.00
crouch walk									
0,2	0.30	0.00	0.20	-0.2	1.10	-2.17	0.62	-0.97	0.44
1,3	fc	2.20	0.00	-0.3	-0.70	-0.05	0.20	-0.92	0.44
scissor hop									
0,2	0.27	0.00	0.77	-0.2	0.70	-0.58	0.20	-0.05	0.09
1,3	fc	0.11	0.01	-1.0	-0.82	-0.27	0.20	-0.10	0.12
backwards leaning backwards walk									
0,2	0.22	0.00	0.28	0.2	0.37	-1.41	0.00	-0.05	0.00
1,3	fc	0.60	0.00	0.3	-0.10	-0.05	0.00	-0.10	0.00
fast run									
0,1	0.15	0.00	0.20	-0.2	1.08	-2.18	0.20	-0.05	0.27
run									
0,2	0.21	0.00	0.20	0.0	0.80	-1.84	0.20	-0.05	0.27
1,3	0.00	0.00	0.20	-0.2	1.08	-2.18	0.20	-0.05	0.27
skipping gait									
0,4	0.19	0.00	0.40	0.0	1.04	-1.75	0.20	-0.19	0.20
1,5	0.12	0.00	0.40	0.0	2.25	-2.18	0.20	-0.05	-1.60
2,6	0.26	0.00	0.04	0.0	2.44	-2.09	0.20	-0.05	0.20
3,7	fc	0.18	0.37	0.0	-0.46	-0.05	0.20	-0.10	0.20
3D walk									
0,2	0.3	0.5	0.2	0	0.5	-1.1	0.6	-0.05	0
lat		0.5	0.2	0	0	0	0	0	0
1,3	fc	0.5	0.2	0	-0.1	-0.05	0.15	-0.1	0
lat		0.5	0.2	0	0	0	0	0	0
3D run									
0,1	0.3	0.5	0.2	0	0.5	-1.1	0.6	-0.05	0
lat		0.5	0.2	0	0	0	0	0	0

**Table 1:** 2D and 3D locomotion parameters for the periodic, left-right symmetric gaits. The columns from left to right represent the state numbers, state dwell duration, position and velocity balance feedback coefficients, and the torso, swing-hip, swing-knee, swing-ankle, stance-knee, and stance-ankle target angles. All angles are expressed in radians. The 2D and 3D runs have only two states.