Data-Driven Biped Control

Yoonsang Lee Sungeun Kim Jehee Lee

Seoul National University

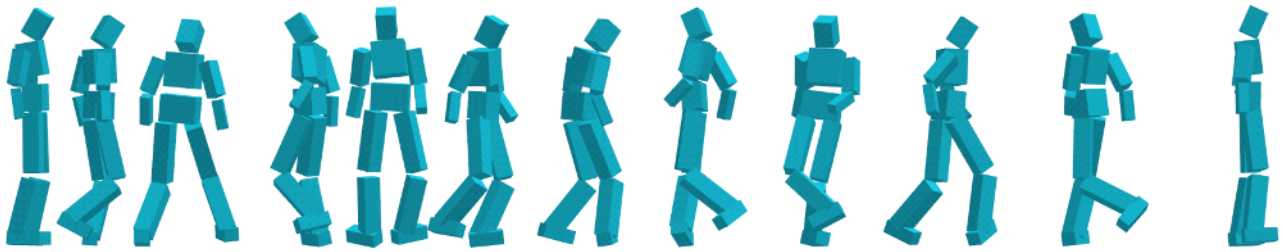


Figure 1: Our data-driven controller allows the physically-simulated biped character to reproduce challenging motor skills captured in motion data.

Abstract

We present a dynamic controller to physically simulate under-actuated three-dimensional full-body biped locomotion. Our data-driven controller takes motion capture reference data to reproduce realistic human locomotion through realtime physically based simulation. The key idea is modulating the reference trajectory continuously and seamlessly such that even a simple dynamic tracking controller can follow the reference trajectory while maintaining its balance. In our framework, biped control can be facilitated by a large array of existing data-driven animation techniques because our controller can take a stream of reference data generated on-the-fly at runtime. We demonstrate the effectiveness of our approach through examples that allow bipeds to turn, spin, and walk while steering its direction interactively.

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

Keywords: Character Animation, Bipedal Locomotion, Physically Based Animation, Motion Capture, Data-Driven Control

1 Introduction

Physically simulating under-actuated biped locomotion has been a notorious challenge in computer graphics for two decades. Most of early biped controllers were either manually designed and hand-tuned or relying on optimization with energy-minimizing objectives. Though some of those controllers are very robust, they tend to result in stereotyped gaits often looking robotic and lifeless. SIMBICON is an exemplar of manually-crafted biped controllers, which is simple, easy-to-implement, and remarkably robust [Yin et al. 2007]. Its robustness allowed it to be employed in further challenges such as controller adaptation [Yin et al. 2008], composition [Coros et al. 2009] and stepping planning [Coros et al. 2008]. However, it is also true that SIMBICON generates stereotyped, marching-like gaits.

Recently, motion capture data were employed to achieve natural and realistic locomotion from physically based controllers. Reference-tracking controllers pose yet other challenges such as collecting physically-feasible training data and developing robust feedback control algorithms for the tracking of reference trajectories and the maintenance of balance.

Our goal is to build full-body, three-dimensional locomotion controllers those are as simple and robust as SIMBICON, and still can faithfully reproduce natural and realistic locomotion guided by reference motion capture data. Our data-driven controllers can generate a variety of locomotor behaviors, such as turning and spinning. The key challenge of our approach is modulating a continuous stream of reference data in a seamless way while synchronizing with forward dynamic simulation. Reference-tracking controllers often fail when the swing foot lands on the ground earlier/later than the reference data indicates. Because the ground reaction force is the only source of control to balance under-actuated bipeds, unexpected changes in ground contacts could easily drive the controllers to unrecoverable failure states. In this paper, we show that carefully synchronizing the reference trajectory and the simulated biped at contact changes in a feedback loop allows us to achieve both the robustness of feedback controllers and the quality of motion capture data simultaneously.

The biggest advantage of our approach is that physically based control can be facilitated by a large array of existing data-driven animation techniques. Our biped controllers are equipped with a data-driven animation engine at the front end. The data-driven engine generates a sequence of movement patterns by editing [Lee and Shin 1999; Kim et al. 2009], blending [Rose et al. 1998], retargetting [Gleicher 1998], and composing [Lee et al. 2002; Kovar et al. 2002] motion fragments in the database. In this framework, the role of dynamic controllers can be greatly simplified, that is, tracking reference trajectories. We will demonstrate the effectiveness of our approach through examples that allow bipeds to turn, spin, and walk while steering its direction interactively.

2 Related Work

Biped controllers have extensively been explored in computer graphics and robotics. Hodgins and her colleagues presented manually designed biped controllers for highly-dynamic athletic motor skills, such as running, jumping, and bicycling [Hodgins et al. 1995]. Their controllers were equipped with finite state machines for phase transition control and a feedback balancing mechanism based on step placements. van de Panne and his colleagues have extensively studied the design of biped and quadruped controllers. Most notably among them, SIMBICON is a robust, three-dimensional walking controller employing a series of key-poses to shape a reference trajectory and a step-based feedback loop to follow the reference trajectory [Yin et al. 2007]. Robust walking controllers can also be acquired by approximating bipeds with simplified inverted pendulums, which allow a guaranteed balancing strategy in a closed-form solution [Kim et al. 2007; Tsai et al. 2009]. Those walking controllers tend to raise their swing feet higher and keep them in the air longer than natural human walking. Their robustness partly comes from the tendency of extended swing phases, which allows more flexibility in step placements for balancing.

Optimization has served as a key methodology in biped controller design. Hodgins and Pollard [1997] adapted existing controllers to new characters of different scales by searching control parameters via simulated annealing. This type of optimization is a very challenging problem because each controller has a lot of parameters to tune and the objectives are highly non-linear. The continuation method employed by Yin et al. [2008] addressed a difficult controller-adaptation problem by solving a progressive sequence of problems that trace a path from a solved problem to the target unsolved problem. Wang et al. [2009] optimized SIMBICON controllers to allow more human-like gaits using biomechanically-motivated objective functions. Control policy searching techniques have also been used to learn walking controllers of physical robots [Morimoto et al. 2003; Tedrake et al. 2004]. It should be noted that our research goal is different than trajectory optimization [Liu and Popović 2002; Fang and Pollard 2003; Safonova et al. 2004; Wampler and Popović 2009], which attempts to find a specific trajectory of motion minimizing energy consumption subject to user constraints.

Once a collection of robust controllers are acquired, high-level control over biped behaviors is desired. Faloutsos and van de Panne [2001] discussed the precondition and postcondition of individual controllers to make transition between controllers. Integrated controllers equipped with various motor skills have been employed to clear stepping stones [Coros et al. 2008] and steer through obstacles [Coros et al. 2009]. da Silva et al. [2009] studied a combination of controllers to create inbetween controllers and coordinate the operation of multiple controllers.

Several biped controllers have been supplemented with the realism of motion capture data. To circumvent the difficulty of balance control, some controllers allow only the upper-body to be driven by motion capture data while the lower-body is either fixed or controlled by a conventional balance controller [Zordan and Hodgins 2002; Nakaoka et al. 2003]. Data-driven control of two-dimensional biped locomotion was first addressed by Sok et al. [2007]. They pointed out that motion capture data are physically inaccurate and rectified motion capture data to make them physically plausible using spacetime optimization. They also demonstrated that even very simple regression and tracking methods can generate stable biped walking, running and jumping when they are combined with physically plausible reference data. da Silva et al. [2008b] developed three-dimensional walking controllers that exhibit improved robustness and stability. Their controller employed short-horizon tracking and quadratic programming to maintain biped balance. This idea has further been improved with LQR (Linear Quadratic Regulator) balance control, which precomputes optimal balancing strategies using a simplified 3-link character model [2008a]. Muico et al. [2009] employed an even more sophisticated model, NQR (Nonlinear Quadratic Regulator), to track the full DOFs of a human body model. Their controllers coped with non-penetration constraints by incorporating an NQR formulation into a linear complementary problem.

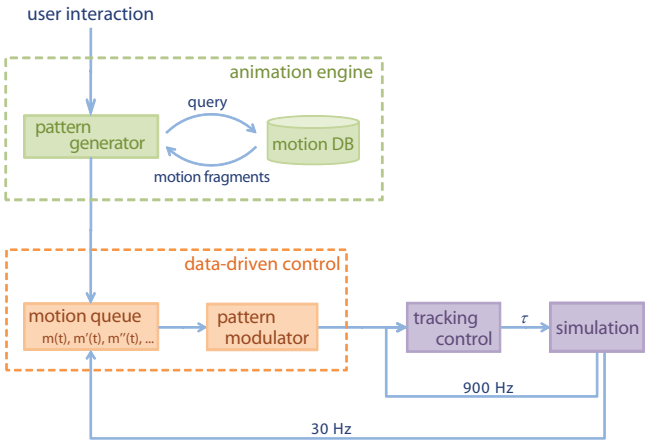


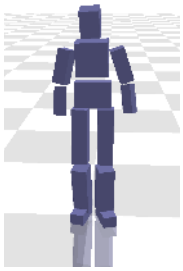
Figure 2: System Overview

In robotics, biped humanoid robots are often driven by oscillatory movement pattern generators described by differential equations. Several researchers have explored a strategy to synchronize reference movement patterns and an actual humanoid via phase resetting [Nakanishi et al. 2004; Nakanishi et al. 2006]. Phase resetting is conceptually similar to our synchronization method, though our controller can deal with realistic motion capture references that necessitate coordinated movements of many actuated joints (upto 42 degrees of freedom in our experiments).

Comparing to previous data-driven controllers, our controller requires neither any precomputed control model (such as regression models, LQR, or NQR) nor non-linear optimization (such as quadratic programming). The model-free approach allows our controllers to take any reference trajectory generated on-the-fly at runtime. It also allows us to decouple the data-driven animation engine and physically based controllers. Therefore, any existing data-driven techniques can be used to actuate and drive physically simulated bipeds without any restriction or precomputation. Our controller does not require the derivative evaluation of equations of motion or a non-linear optimization solver. This makes our controller easy-to-implement and computationally efficient.

3 System Overview

Our interactive biped control system consists of three main components (see Figure 2): Animation engine, data-driven control, and dynamic tracking control. The animation engine provides the user with high-level control over the behavior of the simulated biped through interactive user interfaces and generates a stream of movement patterns by searching through the motion database. The stream of patterns are fed into the reference motion queue and then consumed by tracking control that drives the biped through forward dynamic simulation. The key challenge is with data-driven control, which continuously modulates the reference trajectory such that even a simple tracking controller can reproduce the reference motion. The role of data-driven control is twofold: Maintenance of



	Mass	Length	Inertia (Kg m^2)		
	(kg)	(m)	Lx	Ly	Lz
head	3	0.180	0.012	0.012	0.008
upper arm	2	0.238	0.011	0.011	0.003
lower arm	1	0.200	0.004	0.004	0.001
torso	8	0.230	0.052	0.068	0.049
hip	6	0.180	0.025	0.047	0.040
thigh	5	0.400	0.072	0.072	0.010
shin	5	0.380	0.065	0.065	0.011
foot	2	0.225	0.010	0.010	0.003

Figure 3: The biped dynamic model has 13 body parts connected by 12 ball-and-socket joints.

balance and synchronization between reference data and the actual simulation.

Motion Database. Our dynamic biped model has 13 rigid body parts (head, torso, pelvis, upper arms, lower arms, thighs, shins, and feet) and 12 actuated ball-and-socket joints connecting the body parts (see Figure 3). The total degrees of freedom of the model is 42 including the six degrees of freedom at the unactuated root (pelvis). We annotated motion capture data with ground contact information in a similar way as done by Lee et al. [2002] and then segmented motion data into fragments where ground contact changes. Each fragment contains a half-cycle of locomotion starting from left foot landing to right foot landing or vice versa. Extended double stance phases (e.g., stop to stand still) and flight phases (e.g., broad jump) are also segmented into fragments where double stance/flight begins or terminates. Motion fragments thus obtained are maintained in a directed graph to allow transitioning between them [Lee et al. 2002].

Tracking Control. Tracking control attempts to follow a reference motion trajectory. Our system used a controller similar to Macchietto et al. [2009]. The desired acceleration at any time instance is computed:

$$\ddot{\theta}_{desired} = k_t(\theta_r - \theta) + k_v(\dot{\theta}_r - \dot{\theta}) + \ddot{\theta}_r, \quad (1)$$

where θ_r, $\dot{\theta}_r$ and $\ddot{\theta}_r$ are the joint angles, joint velocities, and joint accelerations, respectively, estimated from the reference motion data by finite differences. Joint torques are computed from the desired joint accelerations using inverse dynamics and then fed into a forward dynamics simulator to actuate the biped. We use a penalty method to compute ground reaction force. This simple tracking control is easy-to-implement and stable with small integration time steps. Note that tracking control operates at the rate of 900 Hz for stability, while data-driven control operates at 30 Hz to match the requirement of visual fidelity (see Figure 2). We used Virtual Physics to solve inverse dynamics and forward dynamics simulation [Kim 2009]. Because our bipeds are under-actuated, we are unable to solve for joint torques those produce desired accelerations at full degrees of freedom via inverse dynamics. The inverse dynamics of an under-actuated system takes the desired accelerations at actuated joints and external forces (including ground reaction force) as input, and produces output torques at actuated joints. The accelerations at unactuated degrees of freedom (in our case, linear and angular accelerations of the root) are passively determined as a result of applying the output torques to actuated joints. Therefore, the unactuated root cannot be directly manipulated through explicit forces/torques, but can only be maneuvered indirectly via harmonious coordination of actuated joints.

4 Data-Driven Control

Controlling a dynamic biped model to imitate biological locomotion captured from a live actor is difficult because of many reasons. At first, the dynamic model has fewer degrees of freedom than the actual human skeleton and idealized ball-and-socket joints are different than human joints. The physical properties, such as mass and inertia, are roughly estimated based on statistical data. Motion capture data include measurement errors in estimating skeletal movements from markers placed on deforming skin. On the other hand, forward dynamics simulation is sensitive to input conditions and external perturbation. Tracking control of under-actuated bipeds is particularly susceptible to even small deviation in ground contact from the reference trajectory.

Our data-driven controller modulates the reference motion capture data actively and continuously at runtime to compensate for the discrepancy between the desired reference motion and the actual simulation of a biped. Specifically, data-driven control modulates lower limbs to actively maintain balance. It also adjusts the timing of motion to synchronize the reference data to the simulation.

4.1 Balancing

Human balance behavior heavily relies on the hip joints and the stance ankle. As pointed out by Wang et al. [2009], the knees are often near-passive throughout the cycle of natural human walking. We apply SIMBICON-style feedback control laws to the hips and stance ankle.

Consider the reference motion at top of the queue at runtime. Let $\mathbf{M}(t) = (\mathbf{v}_0(t), \mathbf{q}_1(t), \mathbf{q}_2(t), \cdots, \mathbf{q}_n(t))$ be a fragment of motion, where $0 \leq t \leq T$ is the index of motion frames, $\mathbf{v}_0 \in \mathbb{R}^3$ and $\mathbf{q}_1 \in \mathbb{S}^3$ are the position and orientation, respectively, of the root, and $\mathbf{q}_k \in \mathbb{S}^3$ for $k > 1$ is the relative orientation of joint k with respective to its parent link.

Motion frames in the queue should be continuously modulated before consumed by tracking control to compensate for possible loss of balance. Let $\mathbf{M}(t_c)$ be the currently referencing motion frame by the tracking controller and $\mathbf{P}$ be the current configuration (pose) of the biped in the simulation. The current pose is supposed to match $\mathbf{M}(t_c)$ in the reference motion, but may deviate in general. For this reason, simply feeding its subsequent frame $\mathbf{M}(t_c + 1)$ to tracking control would not guarantee stable and precise simulation. Instead, we compute an error-compensating, balance-recovering target pose $\hat{\mathbf{P}}$ at every time instance by applying feedback control laws. It guides tracking control to better follow the reference motion. A continuous stream of target poses computed based on motion capture reference data allows our tracking control to be much less stiff than SIMBICON, which uses PD servos with sparse key poses [Yin et al. 2007].

The target pose $\hat{\mathbf{P}}$ is constructed in three steps starting from the corresponding reference frame $\mathbf{M}(t_c+1)$. We first decide its stance hip angle with respect to its pelvis orientation and then elaborate on the swing hip and the stance ankle to yield balance feedback. The swing leg is further adjusted for better tracking.

Feedback on Stance Hip. For an under-actuated system, we do not have a direct control over its root (pelvis), but it can be controlled indirectly by modulating the stance leg. Let $\mathbf{q}_{\text{pelvis}} \in \mathbb{S}^3$ and $\mathbf{q}_{\text{hip}} \in \mathbb{S}^3$ be the orientation of the pelvis and the stance hip with respect to a global, reference coordinate system. We take pelvis orientation $\mathbf{q}_{\text{pelvis}}$ from reference frame $\mathbf{M}(t_c + 1)$ and hip orientation $\mathbf{q}_{\text{hip}}$ from the current configuration $\mathbf{P}$ to compute the desired hip joint angle $\mathbf{q}_d = \mathbf{q}_{\text{pelvis}}^{-1}\mathbf{q}_{\text{hip}}$. The feedback rule to

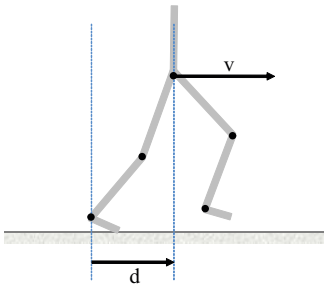


Figure 4: Feedback parameters d and v in the sagittal plane. The center of the pelvis is used as a proxy of the center of mass.

achieve the desired stance hip angle is:

$$\mathbf{q}_{\text{stance\_hip}} = \mathbf{q}_{\text{sth}}\big(\mathbf{q}_{\text{sth}}^{-1}\mathbf{q}_d\big)^{c_0 \cdot s_{\text{stance}}(t)}, \quad (2)$$

where $\mathbf{q}_{\text{stance\_hip}}$ is the stance hip angle of the target pose, $\mathbf{q}_{\text{sth}}$ is the angle of the stance hip at $\mathbf{M}(t_c+1)$, and c_0 is a feedback gain. Transition function $s_{\text{stance}}(t)$ defined over the stance interval allows this feedback control to engage gradually not to make abrupt thrust at the beginning of the stance phase.

Feedback on Swing Hip and Stance Ankle. Our feedback rule on the swing hip is similar to the one of SIMBICON, which monitors the location and velocity of the center of mass (COM) to modulate the swing hip angle. We instead use the relative location and velocity comparing to the reference data. We will explain the feedback rule in two-dimensional sagittal plane for simplicity and clarity. The same procedure should be applied in the coronal plane as well for lateral balancing. Let v and d be the horizontal velocity and location, respectively, of the COM with respect to the stance foot position (see Figure 4). Let v_d and d_d are their desired values estimated from the reference data. Then, the feedback rule is:

$$\theta_{\text{swing\_hip}} = \theta_{\text{swh}} + \big(c_1(v_d - v) + c_2(d_d - d)\big)s_{\text{swing}}(t), \quad (3)$$

where $\theta_{\text{swing\_hip}}$ is the swing hip angle of the target pose, θ_{swh} is the angle of the swing hip at $\mathbf{M}(t_c+1)$, c_1 and c_2 are feedback gains, and transition function $s_{\text{swing}}(t)$ is defined over the swing phase. Similarly, the feedback rule on the stance ankle is defined:

$$\theta_{\text{stance\_ankle}} = \theta_{\text{sta}} + \big(c_3(v_d - v) + c_4(d_d - d)\big)s_{\text{stance}}(t), \quad (4)$$

where $\theta_{\text{stance\_ankle}}$ is the stance ankle angle of the target pose, θ_{sta} is the angle of the stance ankle at $\mathbf{M}(t_c+1)$, and c_3 and c_4 are feedback gains. Intuitively speaking, if the current speed is faster than the reference suggests ($v_d < v$) or the biped leans forwards ($d_d < d$), the biped slows down by extending the stance ankle and landing the swing foot forward farther than the reference trajectory indicates. Conversely, if the current speed is slower than the reference ($v_d > v$) or the biped leans backwards ($d_d > d$), the biped accelerates by bending the stance ankle and landing the swing foot closer.

The sagittal and coronal planes change rapidly for turning and spinning motions and sometimes this can be a source of instability. We used a vertical plane containing a moving direction vector and its perpendicular vertical plane, instead of sagittal and coronal planes, to deal with rapid rotational movements.

Feedback for Swing Foot Height. The simulated biped easily loses its balance when its swing foot mistakenly touches the ground. Our controllers modulates the height of the swing foot from the ground surface with a feedback rule:

$$h_{\text{swing\_height}} = h_{\text{swh}} + \big(c_5(h_d - h) + c_6(\dot{h}_d - \dot{h})\big)s_{\text{swing}}(t), \quad (5)$$

where $h_{\text{swing\_height}}$ is the target height of the swing foot, h and $\dot{h}$ are the current height and its time derivative, h_d and $\dot{h}_d$ are their desired values estimated from the reference data, and c_5 and c_6 are feedback gains. Given the target height, we used an inverse kinematics solver developed by Lee and Shin [1999] to adjust the target pose.

4.2 Synchronization

Let $\mathbf{M}(t)$ be a motion fragment at top of the queue and $\mathbf{M}'(t)$ be its subsequent motion fragment awaiting in the queue. Since we segmented motion data at contact changes, there must be a contact change between the two motion fragments. Typically, for locomotion, a swing foot of $\mathbf{M}$ lands on the ground at the beginning of $\mathbf{M}'$.

In the tracking simulation loop, the swing foot may touch the ground earlier or later than the reference motion indicates even with feedback control. Assume that the tracking controller is currently referencing $\mathbf{M}(t_c)$ when the swing foot is landing. The target pose $\hat{\mathbf{P}}$ computed based on feedback rules may deviate from the reference frame $\mathbf{M}(t_c)$ in general.

If the actual contact was earlier, the remaining frames of $\mathbf{M}$ is dequeued and the next fragment $\mathbf{M}'$ shifts to the top of the queue. At that moment, $\mathbf{M}'$ should be warped to make a smooth transition (see Appendix A for details on mathematical notation).

$$\mathbf{M}(t) \leftarrow \mathbf{M}'(t) \otimes \big(\hat{\mathbf{P}} \oslash \mathbf{M}'(0)\big)^{r(t)} \quad \text{for } \forall t, \quad (6)$$

where $\hat{\mathbf{P}} \oslash \mathbf{M}'(0)$ is the displacement between the two poses. $r(t)$ is a smooth transition function, which is one at the beginning of $\mathbf{M}'$, zero at the end, and its derivatives are zero at both ends (see Appendix B). Intuitively speaking, the displacement at the beginning of $\mathbf{M}'$ propagates gradually to its subsequent frames to make seamless transition. The transitioning period should be as long as possible to achieve smoothest visual transition. One exception is the stance foot, which supports the entire body mass. Even a small deviation at the stance foot may influence the fullbody balance significantly. Transitioning of the stance ankle is handled differently than other joints. At first, a quicker transition of the stance foot usually better stabilizes the next stride. In our experiments, transition function $r(t)$ was set to vary from one to zero over the duration of $\mathbf{M}'$ (usually, a half cycle of locomotion) excepting for the stance ankle, which completes its transition in $1/5$ of the half-cycle duration. Secondly, the angle of the stance foot with respect to the ground surface is more important than tracking the joint angles. Therefore, the target angle of the stance ankle at the end of the transitioning is set such that the angle between the stance foot and the ground surface matches the reference data.

If the actual landing was later than the reference indicates, there are no reference data to follow until the swing foot touches the ground and the next reference motion engages. We expands the current reference motion by integrating joint angles with constant velocities at the end of $\mathbf{M}$ excepting for the stance leg. Similarly expanding the stance-leg motion tends to make it push off the ground with too much thrust. Therefore, we leave the hip, knee and ankle of the stance leg fixed while the reference motion is expanded.

	c1_sag	c1_cor	c2_sag(+)	c2_sag(-)	c2_cor	c3_sag	c3_cor
Walking							
Walk Forward Normal	0.05	0.2	0.2	0.05	0.2	0.1	0.1
Walk Forward Slow	0.05	0.25	0.5	0.05	0.2	0.1	0.1
Walk Forward Fast	0.05	0.3	0.5	0.05	0.2	0.1	0.1
Walk Forward Gentle	0	0.3	1.2	0.05	0.2	0.1	0.1
Walk Wide Swing	0.05	0.2	0.3	0.05	0.2	0.1	0.1
Walk Brisk	0	0.3	0.2	0	0.2	0.1	0.1
Walk March	0	0.3	0.2	0.05	0.2	0.1	0.1
Walk Backward	0.1	0.3	1	0	0.3	0.3	0.3
Spinning, Turning							
Walk Left 45 Degree	0.1	0.4	0.7	0.05	0.2	0.1	0.1
Walk Left 90 Degree	0	0.3	0	0	0.2	0.1	0.1
U-turn	0.05	0.3	1.2	0.05	0.2	0.1	0.1
Spin	0.2	0.2	0.7	0.05	0.2	0.1	0.1
Robustness to Pushes							
Walk Forward Normal	0.05	0.25	1.7	0.1	0.3	0.1	0.1
Interactive Control							
Stop to Walk	0.1	0.25	1	0.05	0.3	0.1	0.1
Normal Walk	0	0.3	1.8	0	0.2	0.1	0.1
Left / Right Turn 90	0.1	0.3	1.5	0.05	0.4	0.1	0.1
Left / Right Turn 135	0	0.3	2	0.05	0.4	0.1	0.1
Left / Right Turn 180	0.1	0.3	2	0.05	0.3	0.1	0.1
Fast Walk	0	0.25	1	0	0.3	0.1	0.1
Walk Style1	0.1	0.4	0.5	0.05	0.4	0.1	0.1
Walk Style2	0.1	0.3	0.5	0.1	0.3	0.1	0.1

Figure 5: Feedback gains are different for sagittal and coronal planes. "+" is for $v_d - v > 0$ or $d_d - d > 0$ and "-" is for $v_d - v < 0$ or $d_d - d < 0$. $c_0 = 1$, $c_4 = 0.1$, $c_5 = 0.5$, and $c_6 = 0.02$ for all examples.

5 Results

All motion data in our experiments were originally captured using a Vicon optical motion capture system at the rate of 120 frames/second and then down-sampled to 30 frames/second. Motion data include walking in a variety of different speeds, turning angles and styles. We also captured sharp U-turning and spinning. We lifted up the swing foot trajectory slightly in the motion data of turning, spinning and interactive control examples which have low step height. We set dynamics and integrator parameters to achieve robust simulation in a conservative manner while maintaining the performance of realtime simulation at the rate of 30 frames/second. The ground reaction is modeled as a damped spring. The ground spring and damping coefficients are $k_s = 2000N/m$ and $k_d = 2\sqrt{k_s} = 89.4Ns/m$, respectively.

Feedback Parameters. All feedback gains are summarized in Figure 5. Parameters were manually tuned for each motion data. Parameter tuning was not formidable because each parameter has an intuitive meaning and many of gains are simply constant for all motion data. Most of motion data can be stably reproduced for a wide range of parameter choices excepting for several very challenging examples, such as spinning, which requires careful parameter tuning (see the accompanied video). Most of our controllers generated stable cycles and became resilient to mild pushes without position feedbacks, that is, $c_2 = c_4 = 0$. Without external perturbation, the velocity feedback alone allows the reference trajectory to be followed closely and thus the position feedback has nothing to do with balancing. However, external pushes would make the simulation to deviate from the reference trajectory and the position feedback would play an important role. Therefore, we first tune c_1 and c_3 to achieve stable cycles without any external perturbation and then tune c_2 and c_4 later in the presence of random pushes at the center of mass.

Locomotion Control. Our biped character is able to reproduce various gaits of human walking (see Figure 6). Each motion clip recorded a subject standing still, starting to walk, taking 6 to 8 steps, and then stopping. Representing motion data as a motion graph allows us to produce an arbitrarily long sequence of locomotion by splicing walking steps. Our biped can track an arbitrary combination of locomotion steps including slow/fast walking, turning of different angles, and different gaits.

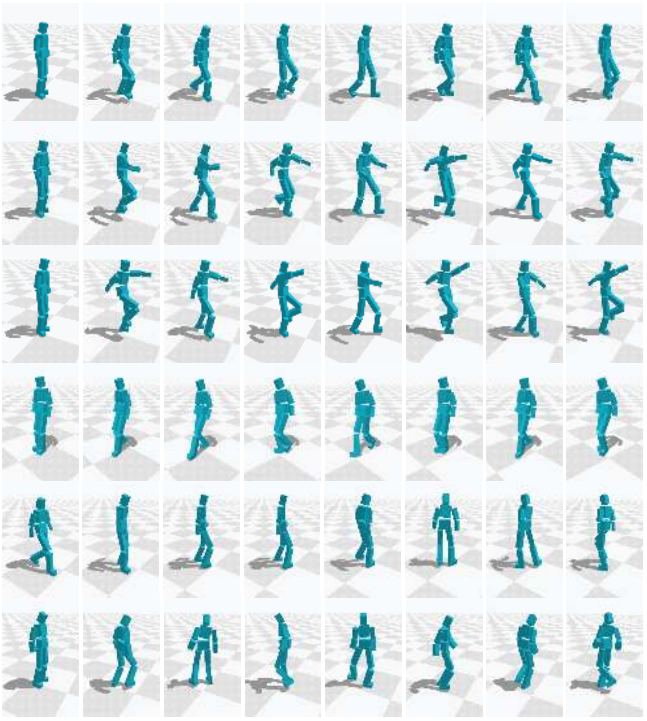


Figure 6: Data-driven biped simulation from motion capture data including (top to bottom) WalkForwardNormal, WalkBrisk, WalkMarch, WalkBackward, U-turn, and Spin.

Effects of Individual Components. Our controller consists of several components for balancing, tracking, and synchronization. Disabling any of these components would result in either falling over in several steps or the degradation in motion quality. We evaluated the effect of each component by disabling each one at a time for a variety of walking data:

- Disabling synchronization always leads to falling over in 3 to 6 steps.
- Disabling the feedback on a stance hip makes the torso to lean and eventually leads to falling over in 2 to 4 steps.
- Disabling the feedback on a swing hip makes the character to lean to one side and eventually leads to falling over in 6 to 10 steps.
- Disabling the feedback on a stance ankle or swing foot height managed to avoid falling over for some gaits, but the motion looks unnatural.

Robustness under Various Conditions. We tested our walking controller on varied simulation conditions to evaluate its robustness (see Figure 7). Our controller generated stable cycles of walking for up to 15 Kg of extra weight on one leg, 50% longer legs, 50% shorter legs, one leg 3% shorter than the other, up and down slopes up to 6 and 4 degrees respectively, 60% to 1200% variations of friction coefficients. These numbers were acquired from the same reference data and the same parameters with those used in walking examples. The limits can be significantly improved if we adapt the

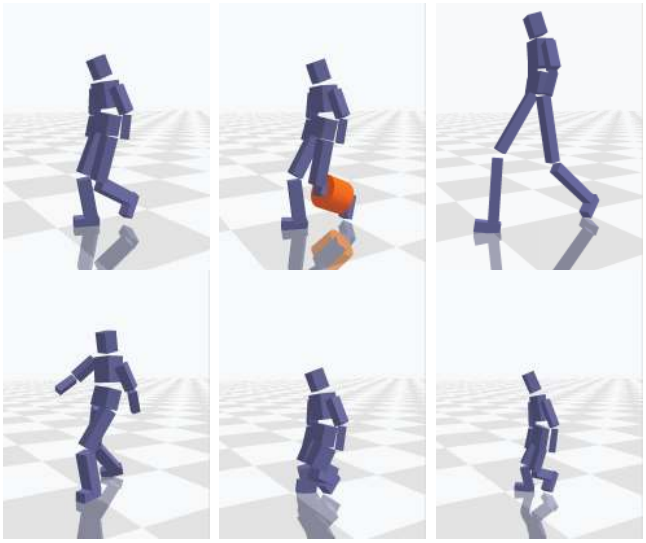


Figure 7: Our walking controller has been tested under varied simulation conditions. (left to right, top to bottom) Original character, extra weight on the leg, longer legs, the left leg shorter than the right leg, shorter height and the same weight, shorter height and lighter weight

reference data kinematically to the varied conditions [Lee and Shin 1999].

Robustness to Pushes. We quantified the robustness to external disturbances with push experiments similar to Wang et al. [2009]. The body mass and simulation coefficients were set to match those of SIMBICON [Yin et al. 2007] as much as we could. However, we were unable to conduct the comparison test under the exactly same condition. Once the biped entered into stable cycles, we applied forces of 0.4 seconds duration to the center of mass of torso once every 4 seconds for 40 seconds. The controller passes the push-resilience test if the biped is still walking stably after 40 seconds. For Walk Forward Normal data, our controller withstands pushes up to 160N, 130N, 80N and 105N from front, rear, right and left, respectively. Because the robustness is influenced by the size and scale of the body, a type of gaits, walking speed, and many other factors, direct comparison to the previous results would be difficult. Roughly speaking, the results indicate that our controller is about as robust as the controller proposed by Wang et al. [2009] and less robust than SIMBICON.

Interactive Control. Our model-free approach allows us to blend a set of motion data on-the-fly and feed inbetween data to the controller. The feedback gains are also interpolated at the same ratio as motion data. The motion set includes turning in 90, 135, and 180 degrees, straight walking at normal/fast speeds and two different gaits. Motion blending and transitioning are computed in the data-driven animation engine and our controller simply tracks a stream of reference data generated at the animation engine. Our biped character can steer in arbitrary turning angles, change its speed, and make transition between different gaits. The user can control the biped character interactively by specifying walking direction, speed and a type of gaits through simple user interfaces. Our controller allows the biped to respond to external perturbations, such as intentional pushes and collision with stacked boxes (see Figure 8).

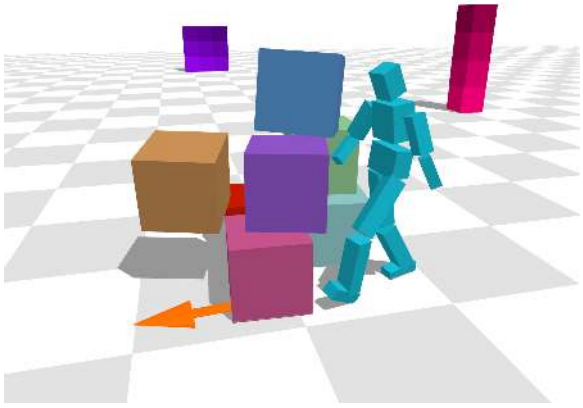


Figure 8: A interactively-controlled biped navigating through stacked boxes.

6 Discussion

Biped control requires two essential mechanisms for shaping trajectories and robust balancing/tracking control. Our work is perhaps emphasizing the importance of trajectory shaping. Motion capture reference data allowed our controller to generate realistic human locomotion. Even balance control was achieved in a data-driven manner by modulating reference trajectories. Presumably, advanced control methodologies would improve our work in several directions. Regression-based approaches [Sok et al. 2007] would allow us to represent natural variations of locomotion in statistical models, which would cope with variations in environments and simulation conditions. Advanced optimal control methods, such as LQR [da Silva et al. 2008a] and NQR [Muico et al. 2009], would allow less stiff systems for tracking control. Even with such expected advantages, we were unable to employ sophisticated control methods because those methods require all reference data be prepared for preprocessing. No reference data generated on-the-fly could be fed into the controller. This restricts the flexibility and versatility of biped control. Designing robust model-free controllers would be an important advance in biped control. We can also think of online model learning that builds a control model incrementally at runtime.

Our controller is more robust if it sacrifices its motion quality by maintaining the direction of the stance foot to match the angle of the ground surface while in contact. It means our controller becomes less robust with more natural stance phases including heel-strike, midstance, and toe-off. In our push experiments, a controller with its stance foot angle fixed with respect to the ground tends to withstand stronger pushes by 10N to 20N. A similar observation was reported by Wang et al. [2009]. The loss of robustness is probably related to inaccurate modeling of the foot. Since our foot model is rigid, it usually have a small contact region on which ground reaction forces are applied. This makes the stance foot to wobble in the simulation. More realistic foot models might improve the robustness of our controller.

Ideally, reference motion data should be physically feasible for best tracking performance, though motion capture data are in general physically imprecise. Some of previous approaches [Sok et al. 2007; Muico et al. 2009] preprocessed motion data to make them physically feasible via spacetime optimization. Spacetime optimization of three-dimensional, full-body motion data is notorious for its challenging nature of numerical instability and heavy computational burden. We did not employ such optimization in our ex-

periments because our feedback rules worked effectively with our test data. However, we suspect that optimized reference data would allow our controller to be more robust and to follow the reference data more precisely. The optimization of reference data would be particularly important if motion data need to be warped, blended, and retargetted.

We have experimented mostly with locomotion data. Controlling and simulating a wider spectrum of human motions will be an exciting avenue for future research. We anticipate that we will see compelling biped characters equipped with a variety of motor skills spanning from low-energy locomotion to highly-dynamic dancing and athletic skills that build on data-driven control techniques such as those presented here.

Acknowledgement

We would like to thank Kyunglyul Hyun and the other members of SNU Movement Research Laboratory for their help in capturing motion data. We are also grateful to Manmyung Kim for his helpful advice about motion processing and Kyungyong Yang for editing the accompanying video. This work was partly supported by MKE/MCST strategic technology development program (Project No. 2008-F-033-02).

References

COROS, S., BEAUDOIN, P., YIN, K. K., AND VAN DE PANNE, M. 2008. Synthesis of constrained walking skills. *ACM Transactions on Graphics (SIGGRAPH Asia) 27*, 6.

COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2009. Robust task-based control policies for physics-based characters. *ACM Transactions on Graphics (SIGGRAPH Asia) 28*, 6.

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

DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Simulation of human motion data using short-horizon model-predictive control. *Computer Graphics Forum (Eurographics 2008) 27*, 2.

DA SILVA, M., DURAND, F., AND POPOVIĆ, J. 2009. Linear bellman combination for control of character animation. *ACM Transactions on Graphics (SIGGRAPH 2009) 28*, 3.

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

FANG, A. C., AND POLLARD, N. S. 2003. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics (SIGGRAPH 2003) 22*, 3, 417–426.

GLEICHER, M. 1998. Retargeting motion to new characters. In *Proceedings of SIGGRAPH 98*, 33–42.

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

HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *Proceedings of SIGGRAPH 95*, 71–78.

KIM, J.-Y., PARK, I.-W., AND OH, J.-H. 2007. Walking control algorithm of biped humanoid robot on uneven and inclined floor. *J. Intelligent and Robotic Systems 48*, 4, 457–484.

KIM, M., HYUN, K. L., KIM, J., AND LEE, J. 2009. Synchronized multi-character motion editing. *ACM Transactions on Graphics (SIGGRAPH 2009) 28*, 3.

KIM, J., 2009. Virtual Physics: The realtime dynamics simulation library, http://virtualphysics.imrc.kist.re.kr/.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics (SIGGRAPH 2002) 21*, 3, 473–482.

LEE, J., AND SHIN, S. Y. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of SIGGRAPH 99*, 39–48.

LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics (SIGGRAPH 2002) 21*, 3, 491–500.

LEE, J. 2008. Representing rotations and orientations in geometric computing. *IEEE Computer Graphics and Applications 28*, 2, 75–83.

LIU, C. K., AND POPOVIĆ, Z. 2002. Synthesis of complex dynamic character motion from simple animations. vol. 21, 408–416.

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

MORIMOTO, J., ZEGLIN, G., AND ATKESON, C. 2003. Minimax differential dynamic programming: Application to a biped walking robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

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

NAKANISHI, J., MORIMOTO, J., ENDO, G., CHENG, G., SCHAAL, S., AND KAWATO, M. 2004. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems 47*, 2-3 (June), 79–91.

NAKANISHI, M., NOMURA, T., AND SATO, S. 2006. Stumbling with optimal phase reset during gait can prevent a humanoid from falling. *Biol. Cybern. 95*, 5, 503–515.

NAKAOKA, S., NAKAZAWA, A., AND YOKOI, K. 2003. Generating whole body motions for a biped humanoid robot from captured human dances. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 3905–3910.

ROSE, C., COHEN, M. F., AND BODENHEIMER, B. 1998. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics & Applications 18*, 5 (September - October), 32–40.

SAFONOVA, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (SIGGRAPH 2004) 23*, 3, 514–521.

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

TEDRAKE, R., ZHANG, T. W., AND SEUNG, H. S. 2004. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2849–2854.

TSAI, Y., LIN, W., CHENG, K. B., LEE, J., AND LEE, T. 2009. Real-Time Physics-Based 3D biped character animation using an inverted pendulum model. *IEEE Transactions on Visualization and Computer Graphics 99*, 2.

WAMPLER, K., AND POPOVIĆ, Z. 2009. Optimal gait and form from animal locomotion. *ACM Transactions on Graphics (SIGGRAPH 2009) 28*, 3.

WANG, J. M., FLEET, D. J., AND HERTZMANN, A. 2009. Optimizing walking controllers. *ACM Transactions on Graphics (SIGGRAPH Asia 2009) 28*, 6.

YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics (SIGGRAPH 2007) 26*, 3.

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

ZORDAN, V. B., AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation*, 89–96.

Appendix A

We represent motion data and motion displacements using mathematical notations of Lee [2008]. Let $\mathbf{P} = (\mathbf{v}_0, \mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n)$ be the pose of an articulated figure, where $\mathbf{v}_0 \in \mathbb{R}^3$ and $\mathbf{q}_1 \in \mathbb{S}^3$ are the position and orientation, respectively, of its root segment and $\mathbf{q}_k \in \mathbb{S}^3$ for $k > 1$ is the relative orientation of joint k with respective to its parent. The displacement between two articulated figure poses can be denoted by an array of linear and angular displacement vectors $\mathbf{D} = (\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_n) \in \mathbb{R}^{3(n+1)}$. Primitive operations between poses and displacements are defined:

$$
\begin{aligned}
\mathbf{P}_1 \otimes \mathbf{P}_2 &= (\mathbf{q}_{1,1}\mathbf{v}_{2,0}\mathbf{q}_{1,1}^{-1} + \mathbf{v}_{1,0}, \mathbf{q}_{1,1}\mathbf{q}_{2,1}, \cdots, \mathbf{q}_{1,n}\mathbf{q}_{2,n}) \\
\mathbf{P}_1 \oslash \mathbf{P}_2 &= (\mathbf{q}_{2,1}^{-1}(\mathbf{v}_{1,0} - \mathbf{v}_{2,0})\mathbf{q}_{2,1}^{-1}, \mathbf{q}_{2,1}^{-1}\mathbf{q}_{1,1}, \cdots, \mathbf{q}_{2,n}^{-1}\mathbf{q}_{1,n}) \\
\mathbf{D}_1 \pm \mathbf{D}_2 &= (\mathbf{u}_{1,0} \pm \mathbf{u}_{2,0}, \cdots, \mathbf{u}_{1,n} \pm \mathbf{u}_{2,n}) \\
\alpha \cdot \mathbf{D} &= (\alpha\mathbf{u}_0, \cdots, \alpha\mathbf{u}_n) \\
\widetilde{\exp}(\mathbf{D}) &= \big(\mathbf{u}_0, \exp(\mathbf{u}_1), \cdots, \exp(\mathbf{u}_n)\big) \\
\widetilde{\log}(\mathbf{P}) &= \big(\mathbf{v}_0, \log(\mathbf{q}_1), \cdots, \log(\mathbf{q}_n)\big) \\
\mathbf{P}^\alpha &= \widetilde{\exp}\big(\alpha\widetilde{\log}(\mathbf{P})\big),
\end{aligned}
$$

where $\alpha \in \mathbb{R}$ is a scalar value, $\mathbf{P}_i = (\mathbf{v}_{i,0}, \mathbf{q}_{i,1}, \cdots, \mathbf{q}_{i,n})$ and $\mathbf{D}_i = (\mathbf{u}_{i,0}, \mathbf{u}_{i,1}, \cdots, \mathbf{u}_{i,n})$. Intuitively speaking, the "difference" between two poses yields displacement $\mathbf{P}_1 \oslash \mathbf{P}_2$. The power $(\mathbf{P}_1 \oslash \mathbf{P}_2)^\alpha$ scales the displacement linearly by a factor of scalar value α. "Adding" the scaled displacement to pose $\mathbf{P}$ yields another pose $\mathbf{P}' = \mathbf{P} \otimes (\mathbf{P}_1 \oslash \mathbf{P}_2)^\alpha$.

Equations (2)-(6) are of an identical form applied to different mathematical objects. Equations (3)-(5) are for scalar values, Equation (2) is for unit quaternions, and Equation (6) is for biped poses.

Appendix B

Transition $s(t) : [a, b] \rightarrow [0, 1]$ is a smooth scalar function that satisfies $s(a) = 0$, $s(b) = 1$, and $s'(a) = s'(b) = 0$. Specifically, we use a cubic polynomial:

$$
\begin{aligned}
s(t) = -2\left(\frac{x-a}{b-a}\right)^3 + 3\left(\frac{x-a}{b-a}\right)^2, &\qquad \text{if } a \le x \le b \\
= 0, &\qquad \text{if } x < a \\
= 1, &\qquad \text{if } x > b.
\end{aligned}
$$

Backward transition function $r(t) : [a, b] \rightarrow [0, 1]$ varies smoothly from one to zero such that $r(a) = 1$, $r(b) = 0$, and $r'(a) = r'(b) = 0$.