Momentum Control for Balance

Adriano Macchietto Victor Zordan Christian R. Shelton

University of California, Riverside

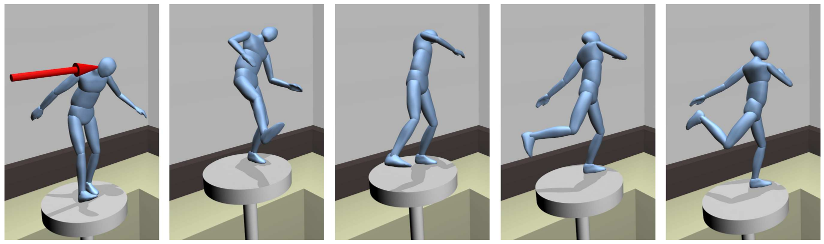

Figure 1: Example response following a disturbance.

Abstract

We demonstrate a real-time simulation system capable of automatically balancing a standing character, while at the same time tracking a reference motion and responding to external perturbations. The system is general to non-human morphologies and results in natural balancing motions employing the entire body (for example, wind-milling). Our novel balance routine seeks to control the linear and angular momenta of the character. We demonstrate how momentum is related to the center of mass and center of pressure of the character and derive control rules to change these centers for balance. The desired momentum changes are reconciled with the objective of tracking the reference motion through an optimization routine which produces target joint accelerations. A hybrid inverse/forward dynamics algorithm determines joint torques based on these joint accelerations and the ground reaction forces. Finally, the joint torques are applied to the free-standing character simulation. We demonstrate results for following both motion capture and keyframe data as well as both human and non-human morphologies in presence of a variety of conditions and disturbances.

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

Keywords: Character animation, physics-based animation

1 Introduction

Control of simulation characters is an important problem in computer animation and has been receiving renewed interest based on recent publication trends. While data-driven techniques have flooded the literature over the past decade, a number of researchers have recently re-focused interest on control techniques for physical models following the lull in such publications since the data-driven animation boom. Most recently, a handful of new motion control approaches have been proposed that can take advantage of the realism of data examples while employing simulation to create characters with movements that are both high quality and can interact in a physically responsive manner [Abe et al. 2007; Sok et al. 2007; Yin et al. 2007; Yin et al. 2008; DaSilva et al. 2008b; DaSilva et al. 2008a]. The cross between the use of motion capture data and physical simulation for characters is revealing a myriad of rich possibilities for improved motion synthesis techniques.

This paper introduces a momentum-based control technique for character simulation. Our method automatically employs full-body balance effects such as the use of arm motion (for example wind-milling) while also resolving conflicting signals between balancing and following a reference motion taken from motion capture or keyframe data. More specifically, our technique guides changes in linear and angular momenta to control the center of mass (CM) and center of pressure (CP) simultaneously. Such momentum control leads to many of the phenomena we commonly associate with whole-body, integrated, and extreme balance activities (see Figure 1). The controller guides a physically based, free-standing character through joint torques computed from desired changes in linear and angular momenta. The resulting simulation is able to retain balance and correct for imbalance in the presence of disturbances and changes in the external environment. In addition to tracking motion capture data for humanlike characters, we demonstrate our approach on characters with unique morphologies. Because we use no heuristics specific to humanoid characters, we can demonstrate the power of our technique trivially on imaginary, multi-limb creatures as well.

The novelty of our approach comes from a set of control laws which dictate appropriate target changes to angular and linear momenta in order to maintain balance. These balance laws specify momentum changes that control the trajectories of the CM and the CP simultaneously. In addition, we present a novel optimization framework which solves for idealized joint accelerations that resolve balance and tracking objectives while constraining the foot to match the acceleration of the ground. These output accelerations are transformed into joint torques using inverse dynamics to maintain the balance of a free-standing simulated character.

2 Background

Generating controllable responsive characters is a challenging open problem in character animation. The goal of generating data-driven, physically simulated characters is shared by several researchers [Zordan and Hodgins 2002; Yin et al. 2003; Abe et al. 2007; Allen et al. 2007; Sok et al. 2007; Yin et al. 2007;

Yin et al. 2008; DaSilva et al. 2008a; DaSilva et al. 2008b] (among others) as well as industry leaders such as Natural Motion (www.naturalmotion.com). The power of these techniques is that they allow the character to react to disturbances through the dynamics, while remaining faithful to a reference motion. Due to competition between controlling the simulation and carefully following the input data, researchers have suggested several alternatives in the form of hybrids kinematic/dynamics models [Shapiro et al. 2003; Mandel 2004; Zordan et al. 2005] and techniques which modify physics-derived parameters, for example [Komura et al. 2004; Komura et al. 2005; Arikan et al. 2005; Yin et al. 2005]. The advantage of such approaches over pure physically based ones is that they do not require sophisticated controllers and problems associated with dynamics, such as balance, become a nonissue. The main limitation of these techniques is that they are not easily generalizable and rely heavily on data for realism. Because they are not physically based, outside of their intended focus they will likely fail in an ungraceful manner. For this reason, we have pursued the physically based motion control approach.

Balance control is an area of interest in several fields including humanoid robotics and character animation. In these fields, where the style of the motion is as important as its effectiveness, often the control problem is framed as one in which a reference trajectory is used to describe the style of a behavior along with corrective activation to maintain an upright stance. Researchers attempt to solve the balance control problem by attending to physical characteristics, such as the center of mass or the zero-moment point (ZMP). However the means by which control has been attempted vary widely, for example, by direct adjustments to the joint angles [Wooten and Hodgins 2000] or, more recently, as a quadratic programming problem which solves for both reference and balance objectives simultaneously [Abe et al. 2007]. This problem is challenging because pre-recorded or pre-generated motions which are desirable for use as reference motions reveal discrepancies between the human (actor) and the simulated character and do not afford environment-specific reactions leading to a host of solutions, for example, the need for motion correction [Sok et al. 2007]. Recently several proposed methods have addressed this question for locomotion [Yin et al. 2007; Yin et al. 2008; DaSilva et al. 2008a; DaSilva et al. 2008b], but less work has focused on balanced standing.

Most researchers have approached balanced standing based solely on control of the CM [Zordan and Hodgins 2002; Abe et al. 2007]. In contrast, the work described in this paper uses momentum control to guide both the CM and the CP which is more robust in extreme balance tasks based on our findings and also leads to the rich appearance of balance motion, especially for the upper body. In closely related work [Kudoh et al. 2002; Kudoh et al. 2006], the ZMP is controlled along with the acceleration of the CM using quadratic programming (QP) to generate responses to large perturbation in standing balance. Their approach is similar to ours in controlling both the ZMP (in our definition, the CP) and the CM. This paper builds on their work with a few key differences. Foremost, they do not precisely control the CP, instead they allow it to move freely within the area of the support polygon. And second, they do not incorporate a reference motion in their QP formulation, leading to responses which do not track a motion capture example, as ours does, and requiring additional (acceleration) constraints, which our technique can ignore. Further, they modulate between two control approaches, stating that their QP method creates large corrections to small disturbances, while our single method unifies all (non-stepping) responses within a single framework. More abstractly, we make explicit the connection between our controlled parameters, CP and CM, and the control of momenta. This distinction is important in that it reveals a symmetry which we exploit in our control laws (described in Section 6).

In robotics, several researchers have recently begun to investigate the potential for angular momentum in balance control, largely for locomotion and stepping [Kajita et al. 2003; Goswami and Kallem 2004; Popovic et al. 2004a; Popovic et al. 2004b; Hofmann et al. 2004]. Goswami and Kallem [2004] support angular momentum as a method for balance with the suggestion that it generalizes other balance concepts such CM control and ZMP maintenance. Popovic and colleagues outline a strong argument for human control of angular momentum and show how it can be regulated for walking movement [Popovic et al. 2004a; Popovic et al. 2004b]. There are also papers in the robotics literature that use angular momentum control for balance, although the proposed control laws vary widely and tend to be simple heuristics crafted for specific effects. Abdallah and Goswami [Abdallah and Goswami 2005] use a simple momentum controller to absorb disturbance effects. Stephens employs a bang-bang control to use the body like a flywheel, applying maximum torque as necessary [Stephens 2007]. And Kajita and colleagues define a control law to set the angular momentum to be zero for control of a humanoid robot [Kajita et al. 2003]. One common theme in all of these robotics papers is that each treats the control of angular momentum as a damper, i.e. to dissipate disturbance. In contrast, we regulate (non-zero) angular momentum to support simultaneous guided control of the CM and CP.

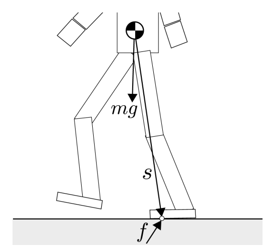

Figure 2: Static force analysis for a standing character.

3 Momentum and the Mechanics of Balance

Basic mechanics shows us that in the absence of external force, the linear and angular momenta of a system, denoted L and H, are conserved. Also, forces and torques applied to the body are equivalent to changes in momenta. Applied to a standing character, if no external perturbations are present, momentum change comes only from the ground reaction forces (GRF) and force due to gravity. We can summarize the momenta/balance relationship simply: assuming a GRF force, f, is applied to the CP at position, p, the linear and angular momentum derivatives are

$$\dot{L} = mg + f \tag{1}$$
$$\dot{H} = s \times f \tag{2}$$

where g is the gravitational acceleration, m is the total mass, c is the CM of the character, and $s = p - c$ (see Figure 2). A simple analysis reveals that controlling linear momentum change is equivalent to controlling the CM acceleration. If we let L_i denote the linear momentum of the ith rigid body, then $L_i = m_i v_i$ where m_i is the mass of body i. And, the momentum of the entire articulated body, L, is computed from the momenta of each individual body, $L = \sum_{i=1}^{n} m_i v_i$ with derivative, $\dot{L} = \sum_{i=1}^{n} m_i a_i$. Or $L = m\dot{c}$ and $\dot{L} = m\ddot{c}$ where $m = \sum_{i=1}^{n} m_i$. From this expression, we find that controlling the derivative of the linear momentum is the same as controlling the mass-scaled CM acceleration. In addition, we can observe from Equation 1 that, in the absence of external forces other

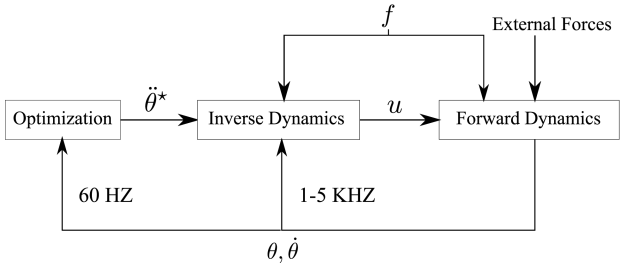

Figure 3: System architecture

than f, if we control $\dot{L}$ with our balancer, we indirectly gain control of the GRF through Equation 1. And, starting from a known state (i.e. a known value for c), Equation 2 completely defines the relationship between the CP and GRF through the change in angular momentum. Thus, using the proposed balance method which controls the desired change in both angular and linear momenta, we gain indirect control over the CM and CP.

The location of the CM projected on the ground plane is a common indicator of the stability of a standing character and, not surprisingly, a common balance strategy is to keep the projection safely within the boundaries of the support polygon. However, control over the CP is also important because (as we can easily see from our analysis) without careful control over the CP, rotation will be induced through angular momentum and the character can tip over. As such, the CP provides a useful measure of the rotational characteristics of the character's state. In addition, if the CP is within the support polygon, where "in" excludes the support polygon boundaries, then the support itself is known not to be rotating. This is important for maintaining balance and, in many balancers that only control the CM, the controller is liable to fail catastrophically when the support begins to tip.

4 System Overview

A diagram of the system components appears in Figure 3. A quadratic optimization with linear constraints is responsible for choosing idealized joint accelerations, $\ddot{\theta}^{\star}$, which meet user-specified goals. These accelerations are handed off to an inverse dynamics module which determines the control inputs, u, in the form of joint torques by solving a hybrid, floating-base algorithm that takes as input the GRF, f and produces physically consistent torques. Finally, a forward simulation component computes the new state based on u, f, and any additional external forces. Note, a similar architecture is presented by Hofmann et al [2004].

Optimization objectives including balance and motion tracking may compete and the optimizer is responsible for choosing the optimal set of accelerations which mutually satisfy each objective. The optimizer solves a quadratic objective function subject to linear constraints. In our system, the number of constraints is low so this optimization can be solved efficiently. The optimization algorithm has no knowledge of the GRF or any other forces except those due to gravity. Instead it relies upon feedback from the forward simulation to correct for disturbances. In the inverse dynamics stage, $\ddot{\theta}^{\star}$ is passed as input along with ground reaction forces, f, to produce the actuator torques which achieve the generalized accelerations in the presence of ground reaction forces. Torque output from the inverse dynamics is fed into the forward dynamics along with the GRF and disturbance forces where it is integrated to produce the final animation. Incorporating the forward dynamics loop allows the system to accurately model external impact dynamics derived from external perturbations. However, the simulation causes a divergence from the idealized accelerations. These errors are corrected in subsequent optimization runs by the feedback components of the tracking and balance objectives.

5 Dynamics

The equations of motion can be written in matrix form:

$$F = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \tag{3}$$

where θ, $\dot{\theta}$, $\ddot{\theta}$ are the generalized coordinates, velocities, and accelerations; F are the generalized forces; M is composed of the inertial coefficients; C represents centripetal and Coriolis components; and G represents the gravitational component. For our system, F has three inputs: control input torques, u; GRF, f; and any additional external forces. Equation 3 can be derived from a Lagrangian formulation of the rigid-body dynamics. The forward dynamics algorithm we use is Featherstone. Featherstone is an efficient $O(n)$ for n bodies, reduced-coordinate algorithm which solves the equations of motion through recursion. In contrast to the competing $O(n^3)$ Composite Rigid Body Method, Featherstone can be shown to be faster when $n > 9$ [Featherstone 2000].

Once the optimization solves for the accelerations, a hybrid, floating-base inverse dynamics algorithm described by Featherstone [1987] is used to convert the accelerations into actuator torques. Unlike the recursive Newton-Euler inverse dynamics algorithm, this algorithm assumes the root is unactuated and generates consistent torques. The inverse dynamics algorithm solves the following equation:

$$u = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) + J^T f \tag{4}$$

for internal joint torques, u, by assuming no externals forces other than GRF, f. Here f represents a vector of Cartesian ground forces and J^T is the transpose of the Jacobian, $J(\theta) = \left[\frac{\partial \theta}{\partial \hat{x}}\right]$ where $\hat{\mathbf{x}}$ is spatial position for the bodies.

In our implementation, a unilateral penalty-based ground contact model is used. Since the forces of penalty methods are based upon the state of the character, and not coupled with the generalized accelerations, penalty forces can be computed and passed into the inverse dynamics algorithm before the forward simulation step. This is in contrast to constraint-based contact models which enforce strict non-penetration constraints and solve for the ground forces and joint accelerations simultaneously as a linear complementary optimization.

6 Control Laws

Our control laws are separated into two balance objectives which attempt to govern the positions of the CP and the projected CM and a tracking objective which attempts to follow a desired motion example. The balance laws dictate desired changes to the two momenta. The tracking law specifies desired accelerations based on the reference motion. Each law is converted into an objective function which is handed off to the optimization to achieve.

6.1 Linear Balance

By maintaining the projection of the CM within the support polygon the character is considered statically balanced. Our goal is to control the trajectory of the CM, for example in order to keep the projected CM close to the center of support. A straightforward rule to control the CM through desired accelerations can be stated as

$$\ddot{c}_{des} = k_l(c_r - c) + d_l(\dot{c}_r - \dot{c}) \tag{5}$$

where k_l and d_l are proportional and derivative gains used to control the acceleration of recovery, and c_r is the reference position for CM, often chosen as the center of the support. $\dot{c}_r$ is the reference velocity of c, usually chosen to be zero. Note, both terms could also be set to follow their respective counterparts in a chosen reference motion, if desired. Equation 5 is only taken along the two orthogonal axes perpendicular to the gravitational axis which lie within the support polygon plane.

Since controlling the derivative of the linear momentum of the character is equivalent to controlling the CM acceleration, $\ddot{c}$, as described in Section 3, we can trivially transform our control law to one which modifies the linear momentum of the character to guide the CM. Through substitution, our linear balance law becomes

$$\dot{L}_{des} = k_l m(c_r - c) - d_l L \tag{6}$$

when we set the desired CM velocity to zero.

For the optimization, the linear momentum objective C_l follows directly.

$$C_l = \|\dot{L}_{des} - \dot{L}\|^2 = \|\dot{L}_{des} - (R\ddot{\theta} + r_{bias})\|^2 \tag{7}$$

where $\dot{L}_{des}$ is computed from Equation 6. The second equation puts the objective in terms of the accelerations which will be determined be the optimization. The calculation of R and r_{bias} can be found in the appendix.

6.2 Angular Balance

We use the angular balance objective to control the CP. An important question is how to control the CP in order to aid in the balance task and not to compete with the linear balance objective. One goal is to ensure the CP does not reach the edge of the support polygon, which could induce support rotation. In addition, we assume preventing the CP from moving too quickly is generally a good choice and will help the linear balancer avoid discontinuous changes in momentum. Starting from these basic heuristics, we derive a control law similar to Equation 5 for the CP, p.

$$\ddot{p}_{des} = k_h(p_r - p) + d_h(\dot{p}_r - \dot{p}) \tag{8}$$

where k_h and d_h are gain constants, p_r and $\dot{p}_r$ are the reference values for the CP and CP velocity, respectively. In our implementation, $\dot{p}$ is determined by finite difference and we found that the center of support is a good choice for p_r. We set $\dot{p}_r$ to zero.

Equation 2 relates the change in angular momentum, $\dot{H}$, with the CP. To remove dependency on the GRF, f, we can reorganize Equation 1 and substitute it in Equation 2. From this, we derive our desired momentum change value as

$$\dot{H}_{des} = (p_{des} - c) \times (\dot{L}_{des} - mg) \tag{9}$$

where p_{des} is the new target location for the CP. We compute this value from the desired acceleration in Equation 8 by integrating it over the timestep. To align the angular with the linear balance term, we take the desired linear momentum derivative, $\dot{L}_{des}$, from Equation 6.

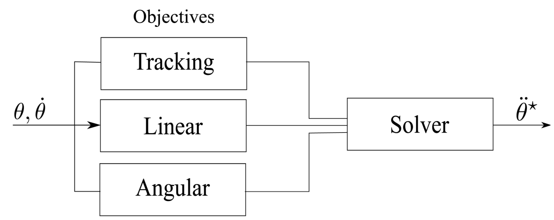

Figure 4: Layout for the optimization block.

As with the linear term, the angular balance objective function for C_h is

$$C_h = \|\dot{H}_{des} - \dot{H}\|^2 = \|\dot{H}_{des} - (S\ddot{\theta} + s_{bias})\|^2 \tag{10}$$

where $\dot{H}_{des}$ is computed from Equation 9. The calculation of S and s_{bias} are also found in the appendix.

6.3 Tracking

Tracking control attempts to follow a prescribed motion trajectory as closely as possible. (In our results, motion data are generated through motion capture and keyframing). Tracking is primarily used to help maintain the stylistic aspects of the desired motion. A controller similar to Abe et al. [2007] is used to provide control:

$$\ddot{\theta}_{des} = k_t(\theta_r - \theta) + d_t(\dot{\theta}_r - \dot{\theta}) + \ddot{\theta}_r \tag{11}$$

where θ_r and $\dot{\theta}_r$ are the motion coordinates and coordinate velocities, and $\ddot{\theta}_r$ is a feedforward acceleration term extracted from the motion data. Introducing the feedforward term allows the feedback tracking gains to be low which allows for less stiff reactions in the presence of external disturbances [Yin et al. 2003]. Our feedforward accelerations are calculated from the reference motion by finite differences.

The tracking objective C_t may now be stated as

$$C_t = \|W_t(\ddot{\theta}_{des} - \ddot{\theta})\|^2. \tag{12}$$

C_t is the sum of the squared errors between the accelerations output from Equation 11 and the accelerations $\ddot{\theta}$ chosen by the optimizer. W_t is a diagonal user-specified weighting matrix which allows for additional tracking emphasis or de-emphasis on particular joints. Thus, a user can create a motion which makes greater utilization of the arms during balance by simply lowering the weights of the corresponding the arm bodies.

7 Optimization

Our optimization routine determines generalized accelerations of the bodies based on the objectives for tracking and balance (See Figure 4). In addition, a small number of equality constraints are set in place in order to prevent the support foot (or feet) from accelerating. The optimization program is stated as such:

$$\begin{aligned} \min_{\ddot{\theta}^{\star}} \quad & \beta_t C_t(\theta,\dot{\theta},\ddot{\theta}) + \beta_l C_l(\theta,\dot{\theta},\ddot{\theta}) + \beta_h C_h(\theta,\dot{\theta},\ddot{\theta}) \\ \text{subject to:} \quad & J_{sup}\ddot{\theta} + \dot{J}_{sup}\dot{\theta} = \hat{\mathbf{a}}_{sup} \end{aligned} \tag{13}$$

where C_t, C_l, C_h represent the tracking, linear and angular balance objective functions of the form $\|W(b - A\ddot{\theta})\|^2$; each β represents the objective weights; and $\hat{\mathbf{a}}_{sup}$ is the spatial accelerations of the supports. Jacobian J_{sup} relates generalized accelerations to the support accelerations.

Our constraint expression ensures that the (foot) supports maintain the linear and angular acceleration of the ground at the point of contact. Let J_{sup} be the submatrix of Jacobian J pertaining to the support. We map generalized velocities to spatial velocities as

$$\hat{\mathbf{v}}_{sup} = J_{sup}(\theta)\dot{\theta} \ . \tag{14}$$

By computing the derivative of Equation 14 over the rows corresponding to our support bodies we obtain the constraint expressed in Equation 13. This constraint is key to our approach because through it the unactuated degrees of freedom of the root are "regained." The basic assumption leading to the form of the constraint is that when the character is statically stable (i.e. the feet are on the ground and the center of mass is within the support polygon) it can realize any acceleration for the root through its contact with the ground. Of course this is limited by acceleration due to gravity, but a maximum upward acceleration equal to (positive) g is fairly conservative for our application. Note, the fixed support constraint does not guarantee that the support will not slip or lift-off when the inverse/forward dynamics phase occurs. The ultimate responsibility of ensuring that the supports remain on the ground lies with our momentum-based balance objectives.

The result of the optimization is a single system of linear equations which can be solved efficiently using any standard matrix-solving algorithm, such as LU decomposition or SVD. Through the objective weights the animator may trade-off between style preservation and balance robustness depending on requirements. Note, while there is some overlap in the structure of our optimizer with the one described by Abe et al. [2007], our formulation is more efficient to execute because we do not use inequality constraints, which requires a more complicated, iterative QP solver. Also, we have many fewer (6) constraints.

8 Extensions

We add several useful extensions to the basic system described.

As we can see in Equation 1, any change in linear momentum must be produced by the GRF. We can indirectly control the GRF applied to the character by controlling $\dot{L}_{des}$, or equivalently, $\ddot{c}_{des}$ in the optimization. In other words, the character may choose to fall with gravity or push towards the ground to increase or decrease the GRF if the freedom exists to do so (i.e. the character can accelerate the CM up or down through the coupling with the ground and gravity). Similarly, if pushed, the character may choose to quickly decelerate the CM resulting in a large tangent force and possible foot slip, or may choose to preserve most of the impact resulting in a small tangential friction force at the expense of the CM potentially accelerating outside of the support polygon. This extension shows how it is possible to extend the linear momentum objective to handle friction and compliance.

Let $N(f)$ denote the magnitude of the normal component of vector f. We choose to clamp the normal component of $\dot{L}_{des}$ to control the GRF such that it remains within a user-specified range:

$$\sigma_l < N(f) = N(\dot{L}_{des} - mg) < \sigma_u \tag{15}$$

where $\sigma_l > 0$, $\sigma_u > N(mg)$ are the lower and upper bounds on the ground normal magnitude. By regulating these bounds a character's conformity with the ground may be intuitively controlled. In practice, we used a range between 0.2 and 1.8 times the character's total weight.

In addition to controlling the normal force of the GRF, we also can control the tangential friction force to keep the supports stationary with respect to translation. Let $T(f)$ denote the magnitude of the projection of f onto the ground tangent plane. We can introduce an additional clamp to ensure that the resulting GRF is within the friction cone:

$$0 < \frac{T(\dot{L}_{des} - mg)}{N(\dot{L}_{des} - mg)} < \mu \tag{16}$$

where μ is the coefficient of friction between the support and ground. To determine the modified $\dot{L}_{des}$, we first clamp the normal component and then clamp the tangential based on the clamped normal component value. This term modifies Equation 6 in the optimization pipeline.

We found better tracking of the reference motion resulted from adding a second optimization loop. The first optimization pass ignores the angular balance objective (i.e. $\beta_h = 0$) and computes a preferred, "ideal" location of the CP, $p^\star$, based on the tracking and linear balance terms alone. In practice, we found that by replacing the value for the CP, p, in the calculation of Equation 8 with this idealized value, $p^\star$, better tracking resulted. Because the first optimization step does not uphold the physical constraints (which is done instead by the forward simulation step) this value can fall outside of the support polygon. Thus, to account for the unrealizable position of $p^\star$, we project its position onto the support area before using it.

It is desirable to control specific points on a body without directly specifying the forward kinematics of the entire character. This problem is analogous to inverse kinematics (IK) where the goal is typically to direct an end-effector to a specified position without deviating too much from a desired posture. Within the proposed optimization framework, soft point acceleration constraints can be implemented as additional objectives. They allow for the optimization to handle multiple, possibly conflicting "constraints" as well as allow for mediating among the optimization objectives. The derivation of the objectives for such point constraints follows succinctly from the definitions for momentum. Note, as with the other objectives, this objective is met (or not) through internal joint torques in the final animation. That is, only the character's internal actuators are used to achieve the point targets. We also implement soft body-orientation constraints in a similar manner.

We found joint limits necessary to prevent the character from moving into impossible postures. We implemented limits using an axis/twist decomposition, where the quaternion representing the joint transformation, q, is decomposed into an axis rotation followed by a twist. For details, see Macchietto [2008] .

9 Implementation and Results

Each simulated actor is composed of n links connected together by 3-DOF actuated ball joints, and a 6-DOF unactuated floating joint connecting the root to the inertial reference frame. All simulations were performed in real-time on a 4200+ AMD Athlon machine. Forward-Euler integration with a step size size of 1-10 khz was used based upon the ground stiffness requirements of the motion. The optimization was recomputed at a separate frequency of 60 hz. Tests were performed across various humanoid and non-humanoid characters for both single-support and double-support motions. All reference motion was generated either by keyframing or motion capture. For the latter, a morphologically-accurate human simulation model was built to match the captured actor. To showcase the robustness of the algorithm to non-humanoid morphologies we also created a four-armed insectoid and a chicken-like character and generated their reference motions using keyframing. All motions were filtered using IK to ensure flat and level support conditions throughout the motion clip. Minimal tuning of optimization parameters was required between clips: the only tuning between characters was the tracking weights, W, to provide greater tracking emphasis on particular body parts.

To test our system, we conducted the following experiments.

Exercise The human performing butterfly and squatting exercises, and a side-kicking motion with various tracking, linear and angular balance objective combinations enabled. The character is unable to complete the motions without the angular balance objective enabled.

Head Drag The human character dragged about by the head using a point constraint while performing repeated squatting motions. The character is able to stay close to the desired motion while attempting to meet the user demands and maintain balance. To avoid joint limitations, the character maneuvers out of the plane in which the point constraint travels.

Cup The human character performs a single-support side kick while holding his head straight and a cup upright. The character manages to perform the side kick while preserving tracking accuracy. Head acceleration tracking improves the posture in comparison to not tracking the acceleration.

Grapple An imaginary creature tracking a keyframed twisting motion is knocked around by a few large impulses (see Figure 5) before being grappled to the ground by multiple user-specified point acceleration constraints. The character adapts the motion gracefully to the multiple constraints without falling. As the creature is "tied down," the motion is adapted into meet these new constraints producing the appearance of a struggle.

Platform A character on a low-friction ($\mu = .1$) moving platform is subjected to multiple external user-specified impulses. The platform is controlled using a sum of sine waves of various frequencies and amplitudes to test the ability of the character to adapt to rapidly changing normal forces. The character adapts to the slippery force while reacting to moderate impacts (See Figure 1).

These experiments (many shown in Figure 6) assessed the characters ability to track the motion accurately while maintaining balance. A morphologically realistic model was used to track captured data for single and double support of a butterfly and squat exercise motion, as well as a single-support martial-arts sidekick motion. We selected motions which required large induced momenta to perform the task well, and the character was able to follow the data faithfully. With only the tracking and linear objectives enabled the character tracked the joint angles accurately; however, due to the modeling error between the actor and the simulation model as well as the inability for the character to control the CP through angular momentum regulation, the system failed to balance. With the angular objective enabled the character was able to remain stable at the cost of a minor reductions in tracking accuracy. Visually, the character still managed to retain the style and accuracy of the original motion. The system also displayed the ability to adapt and track moderately-balanced keyframed motion while upholding overall style. The system had difficulty retaining the style of certain keyframed motions which were too energetic, rhythmic, and unbalanced: as expected, the system would attempt to slow down the motion to retain a desirable CP and the overall style of the motion was lost.

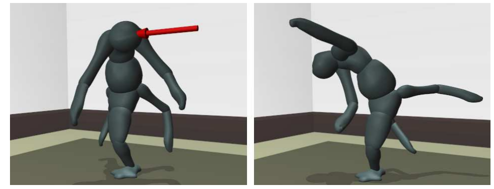

Figure 5: The creature responds to a disturbance in a manner appropriate to its morphology. The unexpected lifting of the lower set of arms is both sensible and adds visual flair to its motion. In a game in which players invented their own creature and a system like the one described here could be used to animate characters in a morphologically consistent manner.

We tested the ability of the character to adapt to a wide-variety of intense environmental conditions involving low-friction, external perturbations, and a moving ground. Extreme balance reactions resulted. While we do not highlight them in the resulting video, not surprisingly the character would topple over if the applied forces were too aggressive. This is to be expected since our balance method cannot change its support foot, for example by taking a step. Yet, the character displayed the agile ability to adapt to the varying ground force magnitudes while reducing relative motion of the support with respect to the platform surface. In addition to retaining stability, the character also displayed natural, lifelike secondary motion not evident in linear-momentum control alone.

We tested the system's ability to adapt to multiple potentially conflicting objectives while simultaneously balancing. In the second row of Figure 6, the character was tested with both orientation and point constraints. Orientation constraints were used to maintain an upright head and cup. In addition, to avoid translational arm jerking, a point constraint with only damping enabled was used to reduce translational arm acceleration. The character completed both tasks while balancing and tracking well without encountering any difficulty in maintaining either head or cup orientation. Figure 6 shows a repeatable behavior in which a point constraint was dragged by the user along the character's sagittal plane. When faced with a near joint-locked scenario, the character gracefully managed to move outside of the plane to find a solution.

10 Discussion

We have presented a novel control routine that employs linear and angular momenta to maintain balance. To control change in momenta, we propose balance laws derived to guide accelerations of the center of mass and center of pressure simultaneously. An optimization acts to turn the objectives into idealized joint accelerations which are, in turn, transformed into joint torques and applied to a full-body simulated character.

Previous approaches have tackled similar problems by incorporating the dynamics and the contact friction cone of the character as an optimization constraint within a quadratic program (QP) formulation [Abe et al. 2007; DaSilva et al. 2008a; DaSilva et al. 2008b]. We attempt to devise a similar solution that is less computationally costly by reducing the guarantee slip-free accelerations. Instead of attempting to optimize over the accelerations, torques, and ground forces simultaneously, we perform an optimization over the accelerations only and rely upon the robustness of our balance objectives to avoid slip. Our argument is that if the character is very close to entering a slip condition, a new behavior controller should be employed, for example to protect from catastrophic failure.

Certain challenges still remain. Due to the stiff penalty-based

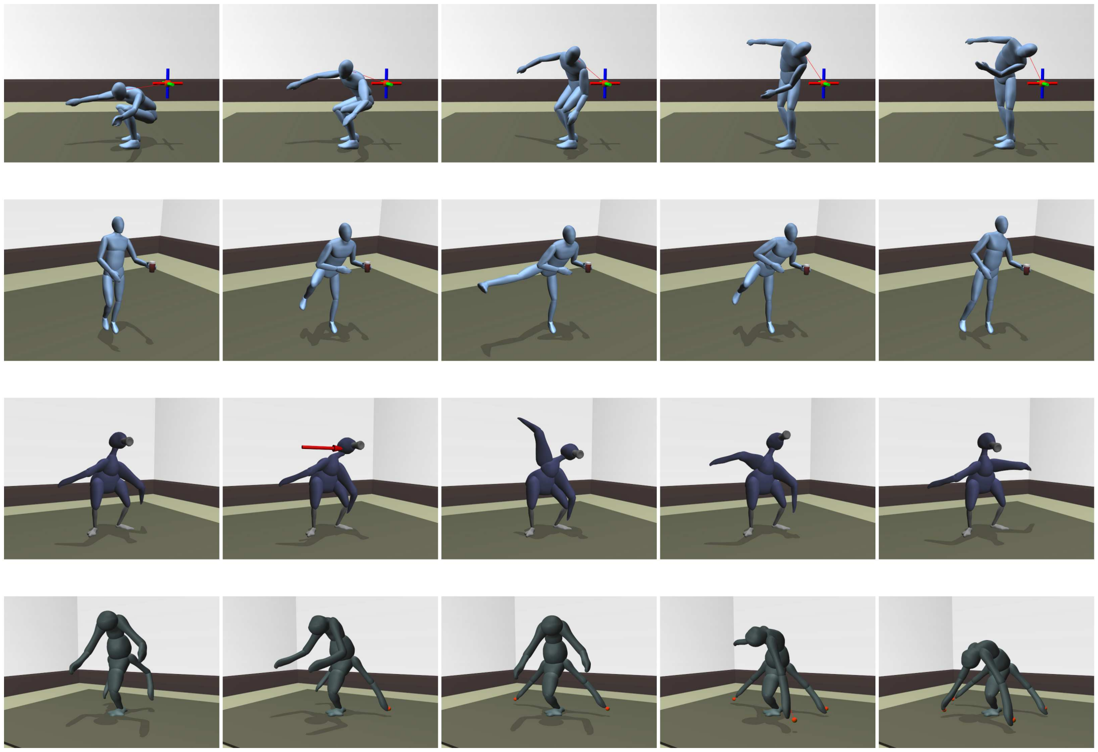

Figure 6: Four filmstrips from the video associated with this paper. On the top we see the character rotating out of the plane of motion as the user guides the character interactively. Next, a side kick is performed while keeping the cup of coffee upright. The bottom two rows show imaginary characters following simple keyframe animation loops while the user interacts with the characters through forces and multiple point constraints. Associated model/parameter files for these animations appear in the auxiliary files for this paper.

ground contact model, small integration steps were required. Future work may involve revising the architecture to incorporate a constraint-based ground contact model. This would require solving a new inverse dynamics problem in which the ground force and actuator torques would need to be solved simultaneously. In addition, compliance is currently the result of clamping the linear momentum control, however resolving the problem of compliance in a more principled manner seems missing. Without compliance the motions were susceptible to teetering and unresponsive to large ground force reactions. And of course the next step is stepping and we anticipate our character simulations will largely benefit from being able to take even a small purposeful step.

In conclusion, this paper presents a unique balance control approach for character animation which uses momenta to drive the CP and the CM simultaneously. Our method achieves desired momenta change via an optimization system that chooses joint accelerations that are held constant at a lower frequency while torques are computed to meet these accelerations in a tight feedback loop using inverse dynamics. We have shown that our characters can remain balancing while following a diverse set of behaviors (keyframed and motion capture), under a wide variety of conditions, while also allowing the character morphology to range from humanlike to imaginary.

A Momentum Derivative Matrices

This appendix shows how the momentum derivative matrices used in Equations 7 and 10 are calculated. We assume that all values are specified in the same frame. This appendix also utilizes the cross product operator $[.]_\times$ which transforms the operand into a 3×3 skew-symmetric matrix which performs a cross product with the multiplicand (i.e. $[u]_\times v = u \times v$) and $\frac{d}{dt}[u]_\times = \left[\frac{du}{dt}\right]_\times$.

For n links, define the following $3 \times 3n$ matrices:

$$T = \begin{bmatrix} M_1 & \ldots & M_n \end{bmatrix} \tag{17}$$

$$U = \begin{bmatrix} m_1[r_1]_\times & \ldots & m_n[r_n]_\times \end{bmatrix} \tag{18}$$

$$V = \begin{bmatrix} I_1 & \ldots & I_n \end{bmatrix} \tag{19}$$

where m_i is the scalar mass, r_i is the vector from the body to the CM, M_i is the 3×3 diagonal link mass matrix, and I_i is the 3×3 inertia matrix of link i computed about the CM of link i. Let

$$P = \begin{bmatrix} T & 0 \\ U & V \end{bmatrix}. \tag{20}$$

The momenta of the entire articulated rigid body may be computed from P and the Jacobian, J:

$$\begin{bmatrix} L \\ H \end{bmatrix} = \begin{bmatrix} \sum_i^n m_i v_i \\ \sum_i^n I_i w_i + r_i \times m_i v_i \end{bmatrix} = PJ\dot{\theta}. \tag{21}$$

Note, the product PJ is denoted the centroidal momentum matrix by [Orin and Goswami 2008] and it is discussed at length in their paper on the topic. Taking the time derivative of Equation 21:

$$\begin{bmatrix} \dot{L} \\ \dot{H} \end{bmatrix} = PJ\ddot{\theta} + (\dot{P}J + P\dot{J})\dot{\theta}\,. \tag{22}$$

Computing $\dot{T}$, $\dot{U}$, and $\dot{V}$ from Equations 17–19 we receive

$$\dot{T} = 0 \tag{23}$$
$$\dot{U} = \begin{bmatrix} m_1[v_1 - \dot{c}]_\times & \ldots & m_n[v_n - \dot{c}]_\times \end{bmatrix} \tag{24}$$
$$\dot{V} = \begin{bmatrix} [w_1]_\times I_1 & \ldots & [w_n]_\times I_n \end{bmatrix}. \tag{25}$$

$\dot{P}$ can now be expressed in terms of $\dot{T}$, $\dot{U}$ and $\dot{V}$:

$$\dot{P} = \begin{bmatrix} \dot{T} & 0 \\ \dot{U} & \dot{V} \end{bmatrix}. \tag{26}$$

R, S, r_{bias}, and s_{bias} presented in Section 8.1 can be specified in the terms discussed as:

$$\begin{bmatrix} R \\ S \end{bmatrix} = PJ \tag{27}$$
$$\begin{bmatrix} r_{bias} \\ s_{bias} \end{bmatrix} = (\dot{P}J + P\dot{J})\dot{\theta}\,. \tag{28}$$

References

ABDALLAH, M., AND GOSWAMI, A. 2005. A biomechanically motivated two-phase strategy for biped upright balance control. In *IEEE Int. Conf. Robotics and Automation*.

ABE, Y., DASILVA, M., AND POPOVIĆ, J. 2007. Multiobjective control with frictional contacts. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

ALLEN, B., CHU, D., SHAPIRO, A., AND FALOUTSOS, P. 2007. On the beat!: timing and tension for dynamic characters. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2005. Pushing people around. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

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

DASILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Simulation of Human Motion Data using Short-Horizon Model-Predictive Control. *Computer Graphics Forum, 27(2)*.

FEATHERSTONE, R. 1987. *Robot Dynamics Algorithm*. Kluwer Academic Publishers.

FEATHERSTONE, R. 2000. Robot dynamics: Equations and algorithms. In *IEEE Int. Conf. Robotics and Automation*.

GOSWAMI, A., AND KALLEM, V. 2004. Rate of change of angular momentum and balance maintenance of biped robots. *IEEE Int. Conf. Robotics and Automation*.

HOFMANN, A., MASSAQUOI, S., POPOVIC, M., AND HERR, H. 2004. A sliding controller for bipedal balancing using integrated movement of contact and non-contact limbs. In *Intelligent Robots and Systems*.

KAJITA, S., KANEHIRO, F., KANEKO, K., FUJIWARA, K., HARADA, K., YOKOI, K., AND HIRUKAWA, H. 2003. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. *Intelligent Robots and Systems*.

KOMURA, T., LEUNG, H., AND KUFFNER, J. 2004. Animating reactive motions for biped locomotion. In *ACM symposium on Virtual Reality Software and Technology*.

KOMURA, T., HO, E. S., AND LAU, R. W. 2005. Animating reactive motion using momentum-based inverse kinematics. *Compute Animation and Virtual Worlds 1*, 16.

KUDOH, S., KOMURA, T., AND IKEUCHI, K. 2002. The dynamic postural adjustment with the quadratic programming method. In *Intelligent Robots and Systems*.

KUDOH, S., KOMURA, T., AND IKEUCHI, K. 2006. Stepping motion for a humanlike character to maintain balance against large perturbations. In *IEEE Int. Conf. Robotics and Automation*.

MACCHIETTO, A., 2008. Momentum-based control for simulated characters. *Master's Thesis, University of California, Riverside*.

MANDEL, M., 2004. Versatile and interactive virtual humans. *Master's Thesis, Carnegie Mellon University*.

ORIN, D., AND GOSWAMI, A. 2008. Centroidal momentum matrix of a humanoid robot: Structure and properties. In *Intelligent Robots and Systems*.

POPOVIC, M., HOFMANN, A., AND HERR, H. 2004. Angular momentum regulation during human walking: biomechanics and control. *IEEE Int. Conf. Robotics and Automation*.

POPOVIC, M., HOFMANN, A., AND HERR, H. 2004. Zero spin angular momentum control: definition and applicability. *IEEE/RAS International Conference on Humanoid Robots*.

SHAPIRO, A., PIGHIN, F., AND FALOUTSOS, P. 2003. Hybrid control for interactive character animation. *Pacific Graphics*.

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

STEPHENS, B. 2007. Humanoid Push Recovery. In *IEEE-RAS International Conference on Humanoid Robots*.

WOOTEN, W., AND HODGINS, J. 2000. Simulating leaping, tumbling, landing and balancing humans. *IEEE Int. Conf. Robotics and Automation*.

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

YIN, K., PAI, D., AND VAN DE PANNE, M. 2005. Data-driven interactive balancing behaviors. *Pacific Graphics*.

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

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

ZORDAN, V. B., AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*.

ZORDAN, V. B., MAJKOWSKA, A., CHIU, B., AND FAST, M. 2005. Dynamic response for motion capture animation. *ACM Transactions on Graphics 24*, 3.