# Control of Rotational Dynamics for Ground and Aerial Behavior

Victor Zordan *UC Riverside*, David Brown *University of British Columbia*,
Adriano Macchietto, KangKang Yin *National University of Singapore*

Simluated rotational behaviors: a parkour style forward roll (top), and a double back handspring (bottom).

***Abstract*—This paper proposes a physics-based framework to control rolling, flipping and other behaviors with significant rotational components. The proposed technique is a general approach for guiding coordinated action that can be layered over existing control architectures through the purposeful regulation of specific whole-body features. Namely, we apply control for rotation through the specification and execution of specific desired 'rotation indices' for whole-body orientation, angular velocity and angular momentum control and highlight the use of the *angular excursion* as a means for whole-body rotation control. We account for the stylistic components of behaviors through reference posture control. The novelty of the described work includes control over behaviors with considerable rotational components, both on the ground and in the air as well as a number of characteristics useful for general control, such as flight planning with inertia modeling, compliant posture tracking, and contact control planning.**

***Index Terms*—Physics-based control, human behavior, torque control, motion capture, character animation**

## I. Introduction

Rotational motion contributes to a wide class of interesting behaviors, including tumbling on the ground, gymnastic actions such as flips and back handsprings, and many martial arts and dance behaviors, for example spinning kicks and break dancing. With the exception of a few cases, such as the work of Wooten and Hodgins [1], most physics-based control papers for character animation have overlooked these types of motions in lieu of other actions, largely including locomotion.

What makes rotation-rich behaviors distinct in terms of control is the need to manage whole-body spin in conjunction with the other aspects of control. That is — in addition to balance, body displacement, and support placement — facing orientation and angular velocity must be guided in meaningful ways to accomplish the proper spin of an action. While this concept can be stated simply, it adds to the complexity in control through the coupling of rotational and translational components both in terms of timing and spatial alignment. While simple feedback may be used, for example, to adjust foot placement in taking a step, there is an inherent assumption that the facing orientation is known and/or close to a known or fixed orientation. However, this assumption becomes invalid when a motion includes a significant rotational component. Thus, in the example of a single flip, the foot placement problem is coupled with a whole-body orientation goal at the same time as the foot is being placed.

Previous techniques sidestepped this issue by discovering specific workable solutions that overcome the issues of this coupling, either through manual tuning [1] or sampling [2]. However, our goal is to address the issue of rotation control in a direct manner. To achieve this goal requires a more general specification of rotational behavior than has been seen to date in computer animation. Further, it raises a number of challenges unique to rotational action. Diverging from non-rotational tasks such as walking, running, and balance, we have identified that rotational behaviors have two aspects that make their control challenging currently:

- Various control *indices* have been proposed for balance and locomotion tasks, e.g., zero moment point (ZMP), but which quantity or quantities to monitor and control during rotational behaviors is not well understood or documented. For example, which guarantee the smoothest progression in a rolling action, or which the most successful initiation and exit of aerial rotation behaviors? General rotational indices have not been proposed, and are not commonly known or discussed. Our investigations in this paper suggest that a collection of rotation indices can be controlled, even synergistically, paralleling observations that no single balance index has been shown to solve all locomotion tasks.
- The contact state and its progression in most locomotion tasks is reasonably clean and phase dependent. Left foot follow right, contact goes from heel to toe, and so on.

However, contact states in rotational behaviors span a large spectrum: from contact-free aerial rotations, such as a flip, to contact-rich ground rotations, such as a sideways (log) roll. To regulate spin successfully across the spectrum of anticipated scenarios, a controller must have access to means for managing control of the character in the absence of contact as well as in the presence of any number of potentially unknown contact configurations.

We propose a hierarchical system that deals with these general rotation control issues through a lower-level local controller which can manage any of a variety of rotational requirements as well as dealing with contact-dependent control issues via supervisory control modules. Further, we add an additional layer of planning to produce signals for specific rotational control indices to produce precision rotation control, including expert gymnastic maneuvers.

Our contributions include:

- The introduction of a systematic study of rotational behaviors and rotation indices, including the introduction of whole-body angular excursion as a rotation control tool;
- The realization of control over these abstract rotation indices through torque regulation using multiobjective optimization;
- The introduction of novel pose-blend tracking and contact-planning mechanisms to aid in control;
- The proposal of an abstract model for automatically producing trajectory-based plans for precision rotational control.

## II. Related Work

Recent publication trends reveal an increased interest in physically based characters. In particular, a host of control techniques have been proposed for behavior activities including running [3], [4], [5], [6], [7], walking [8], [9], [10], [11], [12], [13], leaping [5], and standing (balance) [14], [15]. To date, most full-body control systems have employed some variant of center of mass (CM) control, often coupled with controls or constraints that manage ground contact forces. Several control methods date back to early legged motion controllers, specifically Raibert's control approach for legged robot locomotion [16] which was adapted for general character locomotion in Simbicon-type controllers [9], [11], [12], [17]. These controllers adjust foot placement in order to control CM position and velocity. Other approaches use an abstract CM objective which is realized through optimization to maintain full-body control [14], [10], [15], [18], [5], [6], [7], [13], [19], [20]. However, CM-based control alone cannot account for whole-body rotations in an explicit manner. Methods for controlling large purposeful rotation are lacking.

In this paper, we specifically address control of behaviors with large rotational components such as flips in the air and rolling on the ground. A select number of examples appear in the motion editing literature that consider physical aspects of twisting hops and flipping [21], [22]. However, these methods do not involve forward dynamics simulation and control. A limited number of such control techniques appear for flight-based rotational behaviors [16], [23], [1], [24], [5], although each of these offers little in terms of general methods for rotation control or means of producing new controllers except through manual trial and error. In contrast, our method automatically induces and controls rotation for a wide variety of rotational behaviors in the air, and on the ground.

Control methods for contact-rich rolling motions have been studied even less. The only previous work for humanlike characters to our knowledge is the sampling approach proposed by Liu et al. [2] which demonstrates that open-loop control can be generated (offline) for rolling as well as other rotational behaviors such as cartwheels and kips. However, we target closed-loop controls that can deal with environmental changes interactively. Furthermore, their sampling approach provides little insight about the nature of such rotational behaviors. Liu et al. [25] and Ha et al. [26] show closed-loop control for isolated rolling motion. However, such techniques do not account for perturbations that can accumulate over time (after multiple rolls), indicating a need for more careful control over rotational dynamics. For non-human characters, rolling [27] and flipping [28] have been demonstrated in the context of physics-based simulation and control. These methods rely on modal analysis and user interaction to generate the control respectively, however it is not clear how to generalize these techniques for rotation control of humanoids. For humanlike rolling control specifically, little previous work appears.

## III. Control for rotational behaviors

Many successful locomotion, stepping, and standing control schemes have been proposed. Most such research efforts rely on a *Balance Index* for monitoring and controlling the balance and stability of the character or robots. These indices include center of mass (CM), center of pressure (CP), ZMP, foot rotation index (FRI), centroidal moment pivot (CMP) , CM velocity, the vector from CM to the stance foot or to the CP, as well as linear and angular momenta and their time derivatives [29]. In the course of our investigations in rotation, we found it important to identify of analogous *Rotation Indices* that are useful for controlling and maintaining balance in rotational behaviors.

### A. Rotation indices

**Angular Momentum, AM:** Whole-body angular momentum, $H$, about the center of mass is an obvious choice for rotation control. It is low dimensional and has been shown to be helpful in inducing full-body rotational effects for various activities [15], [5], [20]. However, as a quantity, it is not ideal because it is an aggregate of two factors, whole-body inertia and angular velocity. Further, during the flight phase of aerial (rotating) motions, control over angular momentum is lost. Therefore, while we do exploit angular momentum in our control, we use it primarily as a computed quantity that is driven by our other rotation indices.

**Angular Velocity, AV:** The angular velocity about the center of mass, $\omega$, can be computed as $I_c^{-1}H$ where $I_c$ is the whole body composite inertia matrix about the CM. This value has two features that make it attractive as a control index. First, it

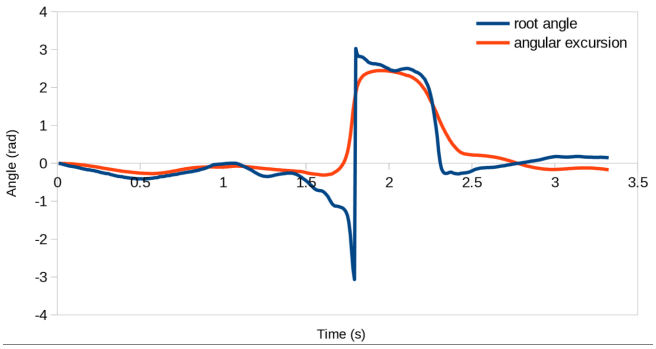



Fig. 1. Off-axis rotation in an aerial data example. Excursion trajectory is much more smooth and well-behaved in contrast to the root. Such characteristics appear in many human motions, including walking.

is fairly intuitive and, second, it can be manipulated when the character is in the air through purposeful shaping of body that leads to change in inertia. Thus, through guiding the body pose, we retain control over the angular velocity during flight. Under these considerations, whole-body angular velocity becomes a powerful choice for general control of rotation.

**Angular Excursion, AE:** Angular velocity itself cannot provide precise control over the absolute orientation of the body. However, to carefully control full-body orientation, for example to finish a flip at the proper angle for landing, requires a robust definition of body rotation. While various surrogates have been used to indicate body rotation in locomotion, such as the root or trunk orientation, large full-body rotations are not well-described by these terms. Instead, we propose the use of the whole-body *angular excursion*, $\phi$, for quantifying the total postural orientation of the character. This value is defined as

$$\phi(t) = \int_0^t \omega \; dt + \phi_0 \tag{1}$$

where $\phi_0$ is a fixed known (zero) reference. Employed by [30] for motion analysis (not control), such whole-body angular excursion may be seen as the angular analog of the CM. We found this value provides a useful handle to regulate full-body orientation of the character. Figure 1 compares the off-axis rotation trajectory of the excursion to that of the root body for a gymnastic aerial.

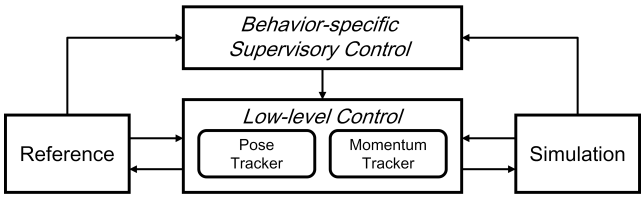



Fig. 2. System Layout

### B. System overview

Our system is derived from a layered hierarchical architecture (Figure 2). At its lowest level, we employ a novel multi-body dynamics module which combines a reduced coordinate dynamics algorithm and semi-implicit force constraint solver (Section IV). We compute activation with a low-level torque controller (Section V) which has the form of previous multi-objective frameworks [14]. Two improved task objectives are proposed, namely momenta trackers (V-A) and a novel pose tracker (V-B). The two provide a means for following reference motion while also controlling based on our described rotation indices. The torque-based tracking controller is guided by behavior specific inputs that are derived via supervisory level controllers, for ground rotation (Section VI) where contact planning is used to decide which collision points to employ in control, and for aerial rotations (Section VII) which employs trajectory optimization for high-level planning.

Successful execution of aerial and fluid progression of ground rotations rely on the exploitation of contact and ground reaction forces (GRFs). For aerial rotations, because of the long flight phase, contact of the supports need to be precisely controlled for successful takeoff and landing. For the ground rotations, an abundance of contacts necessitates a requirement for a selection scheme to discern between solid contacts and incidental ones. Distinguishing between such contacts is critical so that the controller does not attempt to break or prolong contact unnecessarily and can determine which contacts to utilize as the support automatically.

## IV. Constrained Multibody Simulation

We employ a novel semi-implicit simulation which can act stably at large timesteps ($60hz$). This is ideal for character animation, especially for games and interactive settings. Also, we can exploit this formulation for control by informing the controller about current impacts. In contrast, previous methods either ignore such forces or suffer from stability issues that require smaller timesteps. Our proposed technique sidesteps both of these limitations.

Specifically, our simulation extends a known constraint resolution method [31] to work with reduced-coordinate constrained multi-body dynamics system [32]. We use the Featherstone algorithm to handle the body constraints within the character (described as a joint-linked, rigid-body chain). We incorporate Lagrange multipliers to deal with external constraints and kinematic loops between the character and the environment. While Erleben's approach works well for unconstrained rigid bodies, for character animation, articulated motion is required. The reduced-coordinate approach has the benefit of not suffering from constraint-drift between bodies and is therefore well-suited for character systems. The resulting hybrid technique adds clean constraint resolution at large simulation rates without sacrificing quality of the final character motion. Appendix A includes more specifics about the formulation.

## V. Feature-based control

Our controller is formulated as a convex quadratic optimization problem with linear constraints. We solve this problem using quadratic programming (QP) with a feature-control strategy [5]. In this strategy, each abstract objective has the form

$$E(x) = \frac{1}{2}\|Ax - b\|_W^2$$

where $x$ is the optimization state and $A$ and $b$ are a matrix and vector describing linear and constant terms of each objective task function, and $W$ is a weighting matrix that scales the objective error, allowing the user to control the relative importance of each objective within the minimization. The optimization state is the concatenation of the generalized accelerations $\ddot{\theta} \in R^n$, the generalized actuator forces $\tau \in R^{n-6}$, the contact constraint forces $\lambda \in R^{3m}$, and a scalar pose-blending selection parameter, $\gamma$:

$$x = \begin{bmatrix} \ddot{\theta} \\ \tau \\ \lambda \\ \gamma \end{bmatrix}$$

where $n$ and $m$ are the degrees of freedom (DOFs) of the character and the number of contact points, respectively. $\gamma$ is the blend value used as an interpolant weight for blend tracking, described below.Note, only the optimized generalized actuator forces are directly applied to the simulator, the rest of the state is ignored in the simulation update.

In contact, we uphold the zero-work complementarity condition as a heavily weighted objective by enforcing $v_{t+h} = 0$, where $v_{t+h}$ is the post-halfstep velocity (i.e. after velocity integration, but before position integration). We use the post-halfstep velocity, rather than acceleration, to estimate impacts within the control. $v_{t+h}$ is related to the generalized accelerations by the formula:

$$v_{t+h} = J(\theta_t)\dot{\theta}_{t+h} = J(\theta_t)(\dot{\theta}_t + h\ddot{\theta}_t)$$

where $J$ is the constraint Jacobian, and $\dot{\theta}_{t+h} = \dot{\theta}_t + h\ddot{\theta}_t$ by Euler integration rules. In previous work [5], this complementarity condition is not upheld in the control phase, and the effect of the control is compromised by forcing the zero-work condition to be introduced in the simulation. In contrast, we found our approach reduces deviation between the desired outcome of the control and its resulting effect in the simulation. As in similar quadratic control formulations [14], [10], [5], we use a linearization of Coulomb friction, $\|\lambda_T^{(k)}\|_\infty \leq \mu|\lambda_N^{(k)}|$ instead of $\|\lambda_T^{(k)}\|_2 \leq \mu|\lambda_N^{(k)}|$, to keep friction constraints linear. $\lambda_N^{(k)}$ and $\lambda_T^{(k)}$ are the normal and tangent force magnitudes of the $k$'th contact point.

The optimization problem is summarized as follows:

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}\|Ax - b\|_W^2 + \frac{1}{2}\|v_{t+h}\|_{W_v}^2 \\ \text{subject to} \quad & \lambda_N^{(i)} \geq 0, \\ & \|\lambda_T^{(i)}\|_\infty \leq \mu\lambda_N^{(i)}, \qquad i = 1, \ldots, m \\ & S\tau = M\ddot{\theta} + C. \end{aligned} \tag{2}$$

where the last constraint ensures that the generalized forces and accelerations are consistent with the dynamics equations. $M$ is the generalized mass matrix, $C$ are the generalized Coriolis, centrifugal and external forces, and $S$ is a selection matrix that zeros the 6 DOF of the unactuated root. $W_v$ is set to 1000 contrasting unit values in the other objective weights. Both the simulator and control optimization run in lockstep at 60 $hz$.

### A. Momentum tracking

To control linear and angular momenta, $L$ and $H$ respectively, we employ two straightforward objectives that reduce deviation of derivative values from target values $\dot{L}_d$ and $\dot{H}_d$,

$$E_L(x) = \frac{1}{2}\|\dot{L}_d - \dot{L}\|^2$$

$$E_H(x) = \frac{1}{2}\|\dot{H}_d - \dot{H}\|^2$$

To compute the momentum objectives, we specify the respective target values. We determine desired linear momentum change based on CM, $c$, and its derivative. Letting quantities with $\hat{}$ represent the reference/target values, and $K_s$, $K_d$, be the spring and damping gain matrices,

$$\dot{L}_d = m\left(K_s^L(\hat{c} - c) + K_d^L(\dot{\hat{c}} - \dot{c})\right). \tag{3}$$

For rotation control, we introduce an analogous function for angular momentum which controls rotation via two rotation indices, **AE** and **AV**, denoted $\phi$ and $\omega$ respectively.

$$\dot{H}_d = I\left(K_s^H D(\hat{\phi}, \phi) + K_d^H(\hat{\omega} - \omega)\right) \tag{4}$$

where we define function D to be the arithmetic distance operation following [10]. The reference excursion is computed as a discrete sum

$$\hat{\phi} = \prod_{i=1}^{N} \exp(\hat{\omega}_i h)$$

using the exponential operator [33]. Similar for the simulated excursion values.

While such **AM** control has seen some use in recent physics-based animation publications, its control has been used primarily to prevent rotation (e.g. tipping) rather than to induce it. Macchietto et al. [15] employ **AM** regulation to gain control over the center of pressure in balanced standing. Several other researchers [5], [20], [19] follow a zero-spin strategy to damp **AM**, as described in biomechanics for locomotion [30], [34]. One exception is de Lasa et al. [5] where they induce **AM** about the vertical axis to produce a turning jump. In contrast, we employ an abstract angular momentum target value to control rotation through the proposed **AE** and **AV** rotation indices.

### B. Pose tracking

Unlike prior optimization-based controllers [14], [10], [15] that use acceleration-level pose trackers, our *force-based* pose tracker objective takes the form

$$E_P(x) = \frac{1}{2}\|\tau_d - \tau\|^2 \tag{5}$$

where

$$\tau_d = I\left(K_s D(\hat{\theta}, \theta) + K_d(\hat{\dot{\theta}} - \dot{\theta})\right). \quad (6)$$

We found this tracker lead to more stable, compliant pose tracking over one driven by accelerations. Wu and Popović [13] demonstrate similar compliance using "reaction frame" pairs which transmit forces through the character. In contrast, our controller acts at the joint level to produce inter-body response to impacts and unmodeled contacts. Our approach also shares similarities with Tan et al. [35] although the control and testbeds are quite different.

### C. Automatic blend tracking

To make the pose tracking more flexible, we add automatic blend tracking. That is, we provide the system with two reference poses $r_a, r_b$ and allow the control optimization to select $\gamma$, a blend value to track between these poses. The blend tracker objective function has the form

$$E_B(x) = \frac{1}{2}\|\gamma\tau_a + (1-\gamma)\tau_b - \tau\|^2 \quad (7)$$

where $\tau_a$ and $\tau_b$ are computed using Equ. 6 with $\hat{\theta}$ set to $r_a$ and $r_b$ respectively. ($\hat{\dot{\theta}}$ is set to 0.) Through additional constraints added to Equ. 2 to ensure $0 \leq \gamma \leq 1$, this blend tracker allows the optimization to track a linear blend of the desired torques associated with each input pose, without incurring additional cost. Intuitively, this relaxes the demand to track a specific pose and provides the system some freedom over pose selection while maintaining naturalness by staying within the range of the input poses.

For control, we employ this *pose-blend* tracking in two distinct ways. First, we can use extreme poses to sweep out a desired torque space for the given task. For example in rolling, we identify two poses of the desired roll as a tight tuck and fully extended. A noted benefit is that the controller can then automatically select from a meaningful activation range which is defined easily by an animator. Second, we can also use the pose-blend tracker to flexibly follow a reference animation from either motion capture or hand animation. Instead of directly tracking a single pose based on time, we track a blend surrounding a current pose. To do this for rolling, we find the frame of the reference motion that corresponds to the phase of the roll based on the character's angular excursion about the rolling axis, $\phi_r$. We then select the blend poses from the reference motion using a window centered at the found frame's time, $t_r$. Thus, $r_a = r(t_r - \Delta t)$ and $r_b = r(t_r + \Delta t)$ where $r(t)$ is the reference animation and $\Delta t$ is a fixed duration. The blend poses are updated with each optimization pass based on the simulation state. By decoupling the pose selection in this way, the controller has an opportunity to advance or stall the progression of the roll cycle based on the simulation's progress. We found that this approach leads to a more robust roll without deviating greatly from the style embedded in the reference motion.

## VI. Ground contact supervisor

Contact selection control is an issue when various contacts are in flux, as is the case with rolling, because often certain contacts should be employed for control, while others should be ignored. The choice of what contacts to employ for control depends both on the dynamics as well as the strategy. Clearly, humans use specific strategies to induce rolling, for example, it is often desirable *not* to use the head for administering rolling control forces even if the head incidentally comes in contact with the floor. In addition, there are contacts that should not be used for control as they may impede desired strategies intended for the given body part or limb - for example swinging an arm or leg in place for future support. Along with issues, related to strategy, we found a basic need for directing the described controller to avoid enforcing the zero work complementarity condition on contact points unnecessarily. Since our approximation of the zero work condition minimizes contact velocity, the controller will attempt to maintain (stall) the position of all points of contact without discretion. The effect of this is that contact points can become "stuck" by control forces to keep contact when they would otherwise be lifted off. This problem is exacerbated in rolling where contact is highly irregular and many incidental collisions take place that should be ignored entirely by the control routine.

To address this problem, a supervisory routine performs an *inclusion test* to discern which of the contacts the control is to use at a given time step. While search could be employed to explore the optimal contact (e.g. to determine the set of contacts that leads to the best performance), testing every possible contact configuration would be too time consuming. Instead we experimented with several inclusion tests and found one empirically that was suitable for our purposes. Specifically, the contact supervisor performs an initial pass of the control optimization using the full set of contact points, minus contacts on the head, held out for stylistic purposes by default. From this operation, it determines what forces the controller would use naively based on the current conditions of the character. (Note, while we call this naive, it does also embed strategies that might be derived from the reference motion, such as lifting the hands to prepare for a future contact.) It then selects only contacts for which force or torque about the center of mass is above a certain threshold. These contacts are then given to the control optimizer to perform the *actual* optimization. The results of this optimization are used to drive the simulated character. Note, the subset of the contacts is only used for determining the joint torques for control, while the entire set of contacts is always used for ground contact calculations in the forward simulation step. Thus, the supervisor is not producing a change in the physical correctness, only the control strategy surrounding contact.This process is repeated for each time step.

## VII. Flight behavior control

Directly tracking guide signals taken directly from the reference motion usually fails because of the mismatches between the human subject and the virtual character, the differences between the real world physics and the simulated physics, and the unavoidable noise embedded in motion capture trajectories. We therefore need to generate a *viable plan* that is dynamically consistent with our virtual character and

physically realizable in the virtual world. Further, because the low-level controller loses its ability to modify the momenta during flight, careful preparation must be made before take-off to accomplish precise rotation in flight behaviors. For example, in a flip, leaving the ground with enough momentum to make a complete revolution is a requirement. And the choice of how much is 'enough momentum' and how to accomplish this momentum are questions that must be answered for good rotation control over aerial behaviors. Our control for flight relies on a trajectory optimization to provide supervisory control to the low-level controller.

Supervisory control segments flight phases into three stages according to the state of contacts: take-off; flight; and landing (Figure 3). Timing for these stages is derived from the analogous states in the given reference motion. We perform planning through the employment of an abstract model with a state vector, $\mathbf{y}$, that comprises the center of mass $c$; the excursion $\phi$; linear and angular momentum, $L$ and $H$; and two inertial terms, vectors $e$ and $v$:

$$y = \left[c^T, \phi^T, L^T, H^T, e^T, v^T\right]^T.$$

This model is similar to Ye and Liu [20] except that we include the rotation index for excursion, replacing their "integral of angular momentum" term. Also, we add a representation for the composite inertia to handle changing inertia. To represent the inertia, we choose an equimomental ellipsoid, as Lee and Goswami [36], which can be decomposed into an equivalent inertia and vice-versa. $e$ and $v$ represent the radii and rotation vector of this ellipsoid.

**Take-off control** generates a viable trajectory over the time interval immediately preceding lift-off. Reference trajectories $\hat{\omega}(\mathbf{t})$ and $\hat{\dot{\mathbf{c}}}(\mathbf{t})$ are computed through trajectory optimization with the abstract model (Appendix B). The take-off plan is generated through a series of steps — given the current state of the simulation (on the ground) and a specified, desired future state taken from the reference data for the ground phase following the flight. States taken from reference motion are converted to the abstract models form in $\mathbf{y_0}$ and $\mathbf{y_1}$, respectively. Figure 3 is a schematic for the steps of take-off control:

- Working backwards, we first optimize for Stage3, which ends in state $\mathbf{y_1}$. From this, we derive an estimate for the stage $\mathbf{y_b}$.
- Treating $\mathbf{y_b}$ as the final state for Stage2, the controller directly calculates through a ballistic equation the anticipated start state $\mathbf{y_a}$ for Stage2.
- Finally, the trajectory optimizer determines the *viable* plan for taking $\mathbf{y_0}$ to $\mathbf{y_a}$ in Stage 1.

The angular and linear velocity from the trajectory optimization results are fed into the low-level controller while $\hat{\phi}(t)$, $\hat{c}(t)$, and $\hat{\theta}(t)$, are taken from the reference motion. We found that this combination strikes a reasonable trade-off between the requirement to accomplish the behavior and our interest to remain close to the reference.

**Flight control** begins automatically when the predetermined duration for Stage 1 in the reference data has elapsed. The simulation "enters" flight by simply removing the contact forces from the low-level control. In this manner, it is not the literal breaking of contact that constitutes flight, but the release of control over the ground forces. While in the air, we exercise no low-level control other than pose tracking. The end effector which is to act as a support in the next contact stage, however, needs precise control for proper placement to achieve successful landing. So starting from the apex of the flight, the supervisory control computes an optimized kinematic trajectory, described in Appendix C for each end effector. Then we task an acceleration-level feature tracker to move the end effector along this path, through the low-level controller. Such body position and orientation control has been shown an easy extension within the multiobjective framework [15].

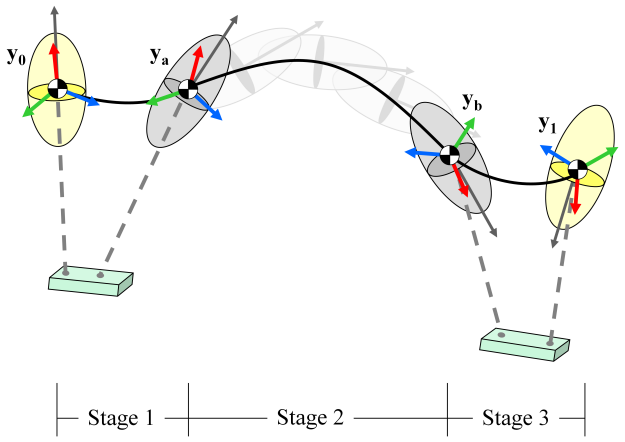



Fig. 3. Take-off control consists of three stages according to the state of contacts: preparation for take-off, flight, and landing. A physically feasible plan for the take-off is computed by looking ahead and working backwards. The landing stage is optimized first based on its desired end state $\mathbf{y_1}$. The planned landing's starting state $\mathbf{y_b}$ becomes the end point for the ballistic flight estimate, where no control is assumed. Finally, the plan for the take-off is optimized to join $\mathbf{y_0}$ to the estimated initial state $\mathbf{y_a}$ of the flight stage.

**Landing control** takes over when the flight stage ends according to reference data. The supervisory control again plans a trajectory on the abstract model, starting from the end state of the flight stage and trying to achieve the desired end state $\mathbf{y_1}$. Note that the end state of the flight stage is from the simulation, not $\mathbf{y_b}$ which is the planned start state in the take-off control. The CM velocity and the angular velocity are again taken from the trajectory planner and fed into the low-level tracking controller. This pass of trajectory planning is similar to the first step of the take-off control. Landing control has been performed successfully for flipping actions previously, by employing a velocity damping phase [23], [1]. Similarly, our system guides the CM velocity, although this velocity is derived automatically via trajectory optimization. Further, we add rotation control through angular velocity, again derived from the trajectory planner. Such rotation control upon landing has not been done to date, to the best of our knowledge.

## VIII. Results

Our results include an array of rotation rich behaviors which are showcased in the supplementary video. In the examples,

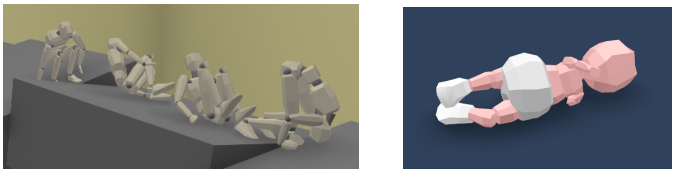


Fig. 4. Forward roll on uneven terrain (left). Baby roll (right).

reference motion (shown in Blue) is shown for comparison as well as used as input to the pose controller.

### A. Ground

We experimented with rolling in various styles and conditions. To control style, we employed reference motions of a single cycle each for behaviors including a basic forward roll, a "parkour" shoulder roll, and a sideways "log" roll. We also employ keyframing to show the strength of our system. For keyframe examples, we hand select a small number of poses in Autodesk's Maya and the output animation motion is treated the same as the human motions. We chose to modify our basic roll cycle to ensure it was symmetric and cycled well, although we found it was unnecessary to do this in general. We employ our automatic pose selection method and manage the indexing of the frames through two processes, one a simple time-dependent playback and the second a phase-locked selection. The latter uses a small fixed duration to select the reference poses based on the roll angle of the characters state. That is, based on the current $\phi_y$ of the character, a small equidistant look-ahead and look-behind from the same excursion value in the reference provides the two poses from which the automatic pose selection acts. Ultimately, control for rolling is derived from the selection of the desired angular velocity and the corresponding linear velocity which are selected once and held fixed for the duration of the behavior. The momentum task objectives uphold these terms even in light of body posture changes (due to pose and contact) which invariably affect the inertia of the character. While we could also induce rotation directly through specifying **AM**, we observe smoother motion by directing the **AV** and (out of plane) **AE** rotation indices.

To test the robustness of our rolling control, we exposed it to various conditions including uneven terrain and environmental hazards, such as shown in Figure 4 left. For rolling along incline and decline slopes we found it necessary to align the linear velocity with the direction of the slope, that is parallel to the slope. Without this, it leads to features such as the character leaping off the ground in a downhill roll. We also show results for a simple form of path following produced by giving small desired deltas to $\phi_z$. In a more whimsical experiment, we conduct a simple task of rolling over for a baby character, as shown in Figure 4 right. This character can be made too weak to accomplish the task and yet the resulting animation reveals a level of determination that communicates the child's intention to roll over, even in the "failed" case. The videos showcase the array of experiments conducted for this behavior control.

### B. Flight

Our results for rotation control include back flip, back handspring, leaping roll, and kip. For each, a single reference motion acts as our starting point. The reference motion is segmented into ground and flight phases and the end states of each flight phase is selected. We convert these into states for the abstract model, and then use these and the trajectory optimizations described in Appendix B to generate the control. Trajectory optimization for each phase requires under one minute of CPU time on a single thread of a Xeon 5600 processor. All other components of the system run at interactive rates, between 15-60 fps depending on the complexity of the model.

The back flip, and the double back handspring as shown in Figure 1, employ the supervisory control for flight behaviors as described in Section VII. A complex behavior such as double back handspring strings together four consecutive flight phases, a series of nine trajectory plans - each feeding one to the next. We show in the accompanying video that turning off the supervisory control of flight planning or turning off the low-level momentum tracking both fail aerial rotational behaviors.

The leaping roll and kip, as shown in Figure 5, have a contact-rich ground rotation phase in addition to the flight phase, and thus the supervisory control method for ground behaviors as described in Section VI is also needed. Our system naturally accommodates control of such hybrid rotations. Currently we manually pick a segmentation point in time to activate the appropriate supervisory control for each phase. The common low-level control for both segments supports seamless transition and integration of the two phases at the simulation level.

Our control system can also generalize the synthesized aerial behaviors beyond what can be observed from the motion capture reference. For example, in one test, we can change the leaping roll to leap down to a lower platform and then continue to roll multiple cycles. In contrast, the reference motion can only leap to the ground of the same height, and roll for just one cycle. In one extra experiment for the kip motion, the character can successfully kip up without the ground rotation control (i.e., using only the flight supervisory control) if we start tracking the motion from when the subject is about to take off.

## IX. Discussion

There are several discussion points that arise from the dynamics and low-level simulation we employ to the specifics

Fig. 5. Leaping roll (top) and kip (bottom).

of our supervisory control. We discuss a select set of these loosely following the progression lowest to highest level.

First, the benefits of our proposed semi-implicit dynamics formulation are that it is both practical for fast simulation and it leads directly to our proposed semi-implicit force-based control technique, which is a desirable improvement for compliance and character control. However, this benefit is only realized by making the low-level optimization run in lockstep with the simulation, and so it can also be seen as a limitation because without this the benefits of the force based tracker break down. Also, we limit ourselves to only static friction in order to handle the zero-work constraints. However, this assumption makes breaking contact more brittle than necessary. We hypothesize that we could soften this limitation by representing the zero-work constratin as a weighted objective in the control, and/or to allow the control to 'break' contact on its own. Ultimately, sliding contact would be a goal for future friction models and, along with it, control under sliding conditions. These issues pose an interesting set of questions related to control over dynamic friction (purposeful slipping).

We experimented with several *inclusion tests* for contact selection before settling upon the one described. Others we tested included a minimal set that gave the controller the ability to generate the necessary force/torque without employing more than was necessary. As well, we tested sets which added biases to contacts lying on the convex hull of all of the contact points. Further, we explored performing inclusion testing following the complete simulation step (and rewinding), with the momenta objectives both turned on and held out. We opted to select our inclusion test because others revealed various visual artifacts and undesirable features. Although we found our solution was suitable for the requirements of ground rotation control and rolling behaviors, we believe more optimal contact selection strategies are likely.

At the behavior level, aerial rotation is observed in the real world to be derived from the combination of the state immediately following take-off and the angular velocities that result from changing the pose in flight. For our control, we assume that the latter matters little in light of the former, since poor take-off conditions cannot be corrected by reasonable changes in the pose; and conversely with good take-off conditions, little is required of an inflight pose controller. Our choice seems justifiable, but an open question becomes apparent, and that is, what can we do with good control over the **AV** via inertia control in the air? Beyond, how important is such control in ground based activities? We make a conscious trade-off in our control structure to manage the **AV** while allowing the pose control to ultimately control the inertia. While this seemed to work well for our experiments with a goal of controlling style, it is unclear that there won't be better solutions ahead that integrate body shape more seamlessly. We plan to explore these topics in future work.

At the supervisory level, we would like to extend our flight planner in the near future to form a type of parametrizable control: we speculate that there is little variance in control across continuous changes in linear and angular velocity in individual motion classes (i.e. where the inertia shape varies little). Using these observations we could reformulate the control optimization to also minimize deviation from known behavior controls to produce realizable "viable" plans in real-time. In addition, the smoothness of the space observed in motion examples we have explored indicates that there is likely a good parametrization of human data which will allow target controls to be constructed, perhaps following models previously proposed for motion data, for example [37], [38].

Parameter tuning in the planning was the most difficult as the choice of segmentation of the motion was done by hand and the flight trajectory optimizer was prone to error propogation that relied on careful selection of timing and good end-effector placement. This was especially true for longer behaviors, such as the double back handspring which includes a chain of four flight plans and end-effector placements. The ground behaviors had fewer sensitive prarameters and we were able to produce many motions with small adjustments to the angular velocity.

Given the lack of examples and the number of reported issues on motions with large rotations found in the latest control results, the exploration of the topic in this paper seems timely and needed. We see this paper not as a conclusory one, but as broaching the subject - which is largely why we highlight the introduction and exploration of possible rotation indices. Our finding on this front is that multiple indices are useful depending on the circumstance, and our observations reveal the potential for more such exploration. Currently, we hypothesize that body-orientation control is a missing component of existing approaches and with even simple control (as we describe for our rolling examples), many frameworks would realize improvement in robustness and visual quality.

## X. Conclusion

As the first effort that systematically studies and synthesizes rotation behaviors, our methodologies provide a solid point of departure for future research effort on similar behaviors. We have not tested skills with a prolonged flight phase, such as platform diving, ski jumps and snowboard stunts [39]. Such rotational performances need more sophisticated inertia shaping mechanisms than we have used in this work. Similarly, we have not tested rotation behaviors around non-principal axes of inertia. Such rotations, however, are not stable and normally do not appear in voluntary rotations.

We have presented a comprehensive system for the simulation and control of rotationally rich behaviors. Rotation behaviors with drastically different contact characteristics including contact-rich ground rotations, contact-free aerial rotations, and hybrid rotations such as a leaping roll that span both extremes, all can be controlled and simulated within the same framework. A collection of rotation indices is proposed and synergistically controlled by our control system. Our constrained multibody dynamics engine is also unique for its stability, compliance compatability, and performance. When put together, the rotation indices and control components enable a powerful and complete system for synthesizing motion skills that have extreme purposeful full-body rotations with built-in physical realism.

## ACKNOWLEDGEMENT

The authors thank Stevie Giovanni and Thai-Duong Hoang for their help in modeling and rendering as well as our reviewers for their helpful and constructive feedback. This work was partially supported by Singapore Ministry of Education Academic Research Fund Tier 2 (MOE2011-T2-2-152) and BCOE initiative funds.


## APPENDIX A CONSTRAINED MULTIBODY DYNAMICS

Our simulator extends the constraint resolution method of Erleben[31] to work with a reduced-coordinate constrained multi-body dynamics system for characters. The linear complementarity problem (LCP) problem of Erleben [31] is given by:

$$\begin{aligned} &u_{t+h} = u_t + hJM^{-1}J^T\lambda - hJM^{-1}f_{ext} \\ &u_{t+h}^{(i)}\lambda^{(i)} = 0 \quad \text{iff } \lambda_{lo} < \lambda^{(i)} < \lambda_{hi} \\ &u_{t+h}^{(i)}\lambda^{(i)} < 0 \quad \text{iff } \lambda^{(i)} = \lambda_{hi}^{(i)} \\ &u_{t+h}^{(i)}\lambda^{(i)} > 0 \quad \text{iff } \lambda^{(i)} = \lambda_{lo}^{(i)} \end{aligned} \tag{8}$$

where $M$ is the $6k \times 6k$ symmetric, positive-definite, block diagonal matrix composed of $k$ $6 \times 6$ rigid-body mass matrix elements; $J$ is the constraint Jacobian which relates change of body coordinates to change in constraint error; $\lambda$, $\lambda_{lo}$, $\lambda_{hi}$ are the constraint force and its bounds; $u_t$ and $u_{t+h}$ are the pre-step and post-step constraint error velocities; $f_{ext}$ are coriolis and external body forces. To add constrained multibody support, we add a generalized $n \times n$ mass matrix for the kinematic chain to $M$ as a block diagonal element and modify the constraint Jacobians to account for the generalized coordinates of the bodies. The LCP is then solved using the iterative projected-gauss solver of Erleben which outputs constraint forces. These constraint forces are fed into the Featherstone forward dynamics algorithm to produce generalized accelerations, and semi-implicit Euler is used to integrate the resulting accelerations to update generalized velocities and positions.

## APPENDIX B TRAJECTORY OPTIMIZATION

The elements of the state $y$ of our abstract model consist of the center of mass $c$; the excursion $\phi$; linear and angular momentum, $L$ and $H$; two inertia vectors, $e$ and $v$:

$$y = \left[c^T, \phi^T, L^T, H^T, e^T, v^T\right]^T.$$

For inertia, we choose to employ an equimomental ellipsoids as Lee and Goswami [36] which we describe by its radii and rotation vector. We assign $e$ and $v$ to define these vectors, respectively.

Our optimization control parameters are the control points for three uniform cubic basis splines, one each for the contact forces, inertial ellipsoid radii and rotation velocities. Let $u_f \in R^{3p_f}$ be the control points for the contact forces, $u_e \in R^{3p_e}$ and $u_v \in R^{3p_v}$ the inertial ellipsoid radii and rotation control points, where $p_f$, $p_e$, and $p_v$ are the number of control points for each spline, respectively. The concatenated control vector is given by:

$$u = \left[u_f^T, u_e^T, u_v^T\right]^T.$$

Let $r_i$ be the position of the $i$th contact point; and $C_f$, $C_e$, $C_v$ be the B-spline functions. The full non-linear system dynamics can now be specified:

$$\begin{aligned} \dot{c} &= m^{-1}L & \dot{\phi} &= W(I^{-1}H) \\ \dot{L} &= \sum_{i=1}^{n} f_i + mg & \dot{H} &= \sum_{i=1}^{n}(r_i - c) \times f_i \\ \dot{e} &= C_e(t,\, u_e) & \dot{v} &= C_v(t,\, u_v) \end{aligned} \tag{9}$$

where $f = \left[f_1^T, f_2^T, \ldots, f_n^T\right]^T = C_f(t, u_f)$. $g$ is the gravity vector. The inertia tensor, $I$, is constructed from the radii and the rotation of the equimomental ellipsoid. Note, any inertia tensor obeying the perpendicular axis theorem can be decomposed into an equivalent equimomental ellipsoid and vice-versa [36]. To represent rotational quantities in the optimization, we use rotation vectors since they do not require normality constraints (as with quaternions) and have continuous derivatives. The system can swap representations as needed, the function $W$ converts an angular velocity to a rotation vector rate of change. We refer the reader to [33] for related details and derivations.

To generate a physically valid segment trajectory we solve a non-linear optimization problem using the Penalty Method which transforms the constrained problem into a series of unconstrained minimization problems. These in turn are solved using Gauss-Newton. All Jacobians required by Gauss-Newton are generated numerically using forward differencing. The trajectory optimization is summarized as follows.

$$\begin{aligned} \min_{y_1, u} \quad & \sum_{j=1}^{m} \|\hat{y}_j - y_j\|_{W_{yj}}^2 + \|\hat{u}_j - u_j\|_{W_u}^2 \\ \text{s.t.} \quad & y_{j+1} = y_j + h\dot{y}_j, \\ & f_{ij}^N \geq 0, \\ & \|f_{ij}^T\|_\infty \leq \mu f_{ij}^N, \\ & c_{min} \leq \|c_j\|_2 \leq c_{max} \quad i = 1, \ldots, n,\ j = 1, \ldots, m \end{aligned} \tag{10}$$

where $m$ is the number of integration steps, $h = T/m$ is the step size and $T$ is the phase length. $\hat{y}_j$ and $\hat{u}_j$ are the desired reference state and control at timestep $j$, and $W_{yj}$, $W_u$ are positive diagonal weighting matrices. We set the target control to be zero to minimize effort. Terms in the objective function attempt to keep the state and control trajectory close to the reference data. The first constraint enforces the dynamics. The second and third constraints ensure the contact forces are within the friction pyramid. The last constraint bounds the distance between the CM and the support assumed to be at the origin for convenience: this helps prevent the optimization from generating a trajectory which is outside the kinematic limits of the character. We choose weights which strongly encourage the optimization to meet the start and end states, since they represent phase transition points that we wish to meet.

**End-effector trajectory optimization.** For each end-effector (EE) that will become a support in the transition from flight to landing, a smooth trajectory is computed using a kinematic trajectory optimization. We represent the EE path as the curve $z$, a uniform cubic B-spline, with control points $u \in R^{3m}$ and the form

$$z = \left[p^T, \phi^T\right]^T = C(u)$$

where $p$ and $\phi$ are the EE's position and rotation vector. We use reference vector $\hat{z} = \left[\hat{z}_1^T, \hat{z}_2^T, ..., \hat{z}_k^T\right]^T$ as a target where: $k$ is the number of trajectory steps; $\hat{z}_1$ is taken from the simulation; $\hat{z}_2, ..., \hat{z}_{k-1}$ are computed from the reference motion; and $\hat{z}_k$ is the final EE state which comes from projecting motion capture reference state to be flat on the ground, so that the EE has a better, solid landing condition. Reference velocity vector, $\dot{\hat{z}}$, has the same construction except that $\dot{\hat{z}}_k = 0$ to ensure the EE hits the ground with zero velocity. We construct the optimization over the control points of the spline as

$$\min_u \sum_{i=1}^{k} \|\hat{z}_k - z(u)\|^2_{W_z} + \|\dot{\hat{z}}_k - \dot{z}(u)\|^2_{W_{\dot{z}}} + \|\ddot{z}(u)\|^2_{W_{\ddot{z}}} \quad (11)$$

where $W_z$, $W_{\dot{z}}$, and $W_{\ddot{z}}$ are positive diagonal weighting matrices which give extremely high weight to meet the initial simulation state, $\hat{z}_1$ and the target resting position for the EE at $\hat{z}_k$ and their derivatives. Like the abstract model trajectory optimization, we employ Gauss-Newton to solve the optimization and all Jacobians are computed numerically.

## REFERENCES


[1] W. Wooten and J. Hodgins, "Simulating leaping, tumbling, landing and balancing humans," in *IEEE Int. Conf. Robotics and Automation*, vol. 1, 2000, pp. 656–662.

[2] L. Liu, K. Yin, M. van de Panne, T. Shao, and W. Xu, "Sampling-based contact-rich motion control," *ACM Trans. Graph.*, vol. 29, no. 4, p. Article 128, 2010.

[3] K. Wampler and Z. Popović, "Optimal gait and form for animal locomotion," *ACM Transactions on Graphics*, vol. 28, no. 3, p. Article 60, 2009.

[4] S. Coros, P. Beaudoin, and M. van de Panne, "Generalized biped walking control," *ACM Trans. Graph.*, vol. 29, no. 4, p. Article 130, 2010.

[5] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Trans. Graph.*, vol. 29, no. 4, p. Article 131, 2010.

[6] T. Kwon and J. Hodgins, "Control systems for human running using an inverted pendulum model and a reference motion capture sequence," in *ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2010, pp. 129–138.

[7] I. Mordatch, M. de Lasa, and A. Hertzmann, "Robust physics-based locomotion using low-dimensional planning," *ACM Trans. Graph.*, vol. 29, no. 4, p. Article 71, 2010.

[8] K. W. Sok, M. Kim, and J. Lee, "Simulating biped behaviors from human motion data," *ACM Trans. Graph.*, vol. 26, no. 3, p. Article 107, 2007.

[9] K. Yin, K. Loken, and M. van de Panne, "Simbicon: Simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, no. 3, p. Article 105, 2007.

[10] M. da Silva, Y. Abe, and J. Popović, "Interactive simulation of stylized human locomotion," *ACM Trans. Graph.*, vol. 27, no. 3, p. Article 82, 2008.

[11] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Optimizing walking controllers," *ACM Trans. Graph.*, vol. 28, no. 5, p. Article 168, 2009.

[12] Y. Lee, S. Kim, and J. Lee, "Data-driven biped control," *ACM Trans. Graph.*, vol. 29, no. 4, p. Article 129, 2010.

[13] J.-C. Wu and Z. Popović, "Terrain-adaptive bipedal locomotion control," *ACM Trans. Graph*, vol. 29, no. 4, p. Article 72, 2010.

[14] Y. Abe, M. DaSilva, and J. Popović, "Multiobjective control with frictional contacts," *ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 249–258, 2007.

[15] A. Macchietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," *ACM Trans. Graph.*, vol. 28, no. 3, p. Article 80, 2009.

[16] M. H. Raibert and J. K. Hodgins, "Animation of dynamic legged locomotion," in *Proceedings of SIGGRAPH '91*, 1991, pp. 349–358.

[17] S. Coros, A. Karpathy, B. Jones, L. Reveret, and M. van de Panne, "Locomotion skills for simulated quadrupeds," *ACM Trans. Graph.*, vol. 30, no. 4, p. Article 59, 2011.

[18] U. Muico, Y. Lee, J. Popović, and Z. Popović, "Contact-aware nonlinear control of dynamic characters," *ACM Trans. Graph.*, vol. 28, no. 3, 2009.

[19] C.-C. Wu and V. Zordan, "Goal-directed stepping with momentum control," in *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2010, pp. 113–118.

[20] Y. Ye and C. K. Liu, "Optimal feedback control for character animation using an abstract model," *ACM Trans. Graph.*, vol. 29, no. 4, p. Article 74, 2010.

[21] Y. Abe, C. K. Liu, and Z. Popović, "Momentum-based parameterization of dynamic character motion," in *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2004, pp. 173–182.

[22] A. Majkowska and P. Faloutsos, "Flipping with physics: motion editing for acrobatics," in *ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007, pp. 35–44.

[23] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien, "Animating human athletics," in *Proceedings of SIGGRAPH 1995*, 1995, pp. 71–78.

[24] M. da Silva, F. Durand, and J. Popović, "Linear bellman combination for control of character animation," *ACM Trans. Graph.*, vol. 28, no. 3, p. Article 82, 2009.

[25] L. Liu, K. Yin, M. van de Panne, and B. Guo, "Terrain runner: Control, parameterization, composition, and planning for highly dynamic motions," *ACM Trans. Graph*, vol. 31, no. 6, p. Article 154, 2012.

[26] S. Ha, Y. Ye, and C. K. Liu, "Falling and landing motion control for character animation," *ACM Transactions on Graphics*, vol. 31, no. 6, p. Article 155, 2012.

[27] P. G. Kry, L. Reveret, F. Faure, and M.-P. Cani, "Modal locomotion: Animating virtual characters with natural vibrations," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 289–298, 2009.

[28] J. Kim and N. S. Pollard, "Direct control of simulated nonhuman characters," *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 56–65, 2011.

[29] M. Popovic, A. Goswami, and H. Herr, "Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications," *The International Journal of Robotics Research*, vol. 24, no. 12, p. 1013, 2005.

[30] M. Popovic, A. Hofmann, and H. Herr, "Angular momentum regulation during human walking: biomechanics and control," in *IEEE Int. Conf. Robotics and Automation*, vol. 3, 2004, pp. 2405–2411.

[31] K. Erleben, "Velocity-based shock propagation for multibody dynamics animation," *ACM Trans. Graph.*, vol. 26, p. Article 12, 2007.

[32] R. Featherstone, *Robot Dynamics Algorithm*. Kluwer Academic Publishers, 1987.

[33] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," 2006, stanford University Tech. Report.

[34] M. Popovic, A. Hofmann, and H. Herr, "Zero spin angular momentum control: definition and applicability," *IEEE/RAS International Conference on Humanoid Robots*, pp. 478–493, 2004.

[35] J. Tan, C. K. Liu, and G. Turk, "Stable proportional-derivative controllers," *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 34–44, 2011.

[36] S.-H. Lee and A. Goswami, "Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots," in *IEEE Intern. Conf. on Robotics and Automation*, 2007, pp. 4667–4672.

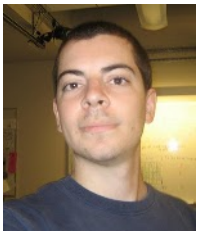
[37] L. Kovar and M. Gleicher, "Automated extraction and parameterization of motions in large data sets," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 559–568, 2004.

[38] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic, "Style-based inverse kinematics," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 522–531, 2004.

[39] P. Zhao and M. van de Panne, "User interfaces for interactive control of physics-based 3D characters," in *ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games*, 2005, pp. 87–94.

**Victor Zordan** is an associate professor at University of California Riverside. His research interests center around game and special effects animation for characters with an emphasis on realism and controllability. Victor has developed a host of techniques that merge motion capture and dynamic simulation to create flexible and responsive behavior for humanoids. His research interests span the field of computer animation with a strong bias in physics-based modeling and interfaces.

**David Brown** is an animation software engineer at 2K Games. He received his B.S. and M.S. in Computer Science from the University of California, Riverside where he worked with Dr. Victor Zordan. He started his PhD at the University of British Columbia with Dr. Michiel van de Panne before joining 2K Games in 2014. He has published several works on character animation and control. His research interests are in developing more realistic and effective methods for controlling virtual characters with a focus on physics based control.



**Adriano Macchietto** received his B.S. and M.S. in Computer Science from the University of California, Riverside. His published research has centered around the topic of momentum control for character physics. Adriano is broadly interested in character animation, game physics, AI, and graphics programming with a focus on creating accurate visual-driven simulations of real world environments.

**Kang Kang Yin** received the PhD degree from the University of British Columbia. She worked as an Associate Researcher at Microsoft Research Asia from 2008 to 2010. She is currently an endowed Assistant Professor in Computer Science at the National University of Singapore. Her research interests include computer animation, geometry processing, and human computer interaction. She is a member of the IEEE and ACM.