Anticipatory Balance Control

Amir H. Rabbani*
McGill University

Michiel van de Panne†
University of British Columbia

Paul G. Kry‡
McGill University

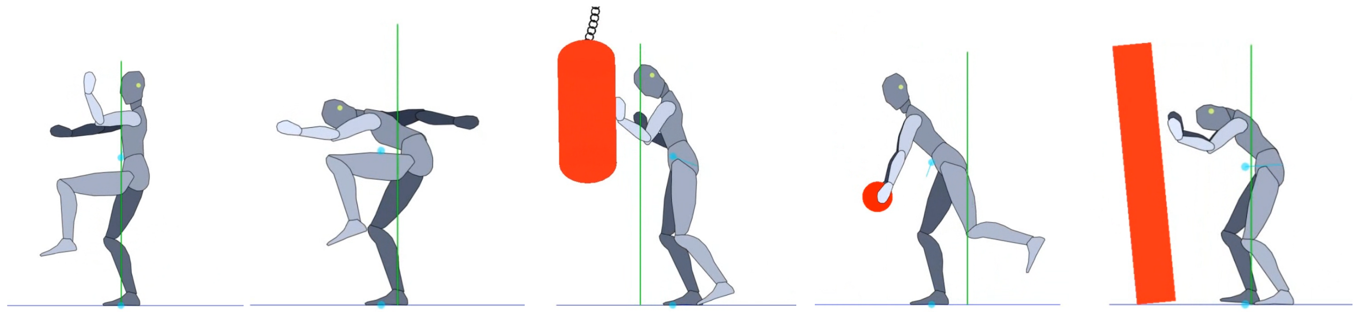

Figure 1: *Learned center-of-mass reference trajectories enable anticipatory control for a variety of tasks, including (from left to right): fast changes of pose; punching; catching and lifting; and pushing.*

Abstract

A hallmark of many skilled motions is the anticipatory nature of the balance-related adjustments that happen *in preparation for* the expected evolution of forces during the motion. This can shape simulated and animated motions in subtle-but-important ways, help lend physical credence to the motion, and help signal the character's intent. In this paper, we investigate how center of mass reference trajectories (CMRTs) can be learned in order to achieve anticipatory balance control with a state-of-the-art reactive balancing system. This enables the design of physics-based motion simulations that involve fast pose transitions as well as force-based interactions with the environment, such as punches, pushes, and catching heavy objects. We demonstrate the results on planar human models, and show that CMRTs can generalize across parameterized versions of a motion. We illustrate that they are also effective at conveying a mismatch between a character's expectations and reality, e.g., thinking that an object is heavier than it is.

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

Keywords: physics-based simulation, computer animation, humanoid balance, momentum control

1 Introduction

The adept control of balance is second nature to humans. However, it is surprisingly difficult to model in simulation. Good solutions would enable the much broader adoption of physics-based character simulations, given the many nuances and interactions that are effortlessly achieved using simulation and which are difficult to achieve using other means. Research on balance strategies over the past two decades has largely focused on the problems of achieving robust standing balance and that of robust locomotion.

Balance controllers have been developed around a variety of principles, including linear momentum control, angular momentum control, model-predictive control, and virtual model control. They vary greatly in the assumptions about the nature of the balance or movement tasks, the abstractions used, and their reactive or anticipative nature.

In this paper we develop a method for learning anticipatory balance strategies that allows for highly dynamic motions, such as fast pose transitions, as well as dynamic interactions with the environment, such as what would occur during a strong push or kick. The key to the approach lies with optimizing the center-of-mass reference trajectory (CMRT) in a way which allows the center of mass to be used in task-specific or motion-specific ways. For example, the character can fall forward to give a large push to an object, knowing that the reaction to the push will then also result in balance being restored. While previous methods allow for the use of CMRTs, this component of the control is often assigned a fixed default value, such as keeping the CM at the center of the support polygon, or they are initialized from motion capture data. Thus, no practical mechanisms are provided for creating suitable CMRTs for new motions that users may wish to author.

We show how the CMRT can be optimized to guide a robust linear-and-angular momentum balance controller towards achieving several types of anticipatory behavior. We demonstrate that CMRTs can be interpolated to achieve anticipatory motions across a parameterized family of motions. A mismatch between the character's expectation and reality is also effectively conveyed using the anticipatory balance strategies. All our results are for planar motions, although we expect that they will generalize well to 3D motions.

2 Related Work

The control of balance has been a topic of interest in computer animation, robotics, and biomechanics for several decades. In this review, we touch on the most closely related work and in particular work related to animation and physics-based character simulation.

*e-mail: amir.rabbani@mail.mcgill.com
†e-mail:van@cs.ubc.ca
‡e-mail:kry@cs.mcgill.ca

Linear momentum regulation is a simple and prevalent strategy for balance control. In its simplest form, this can take the form of a virtual spring-and-damper to compute the force required to pull the center of mass to a desired position and velocity. This force can then be realized using internal joint torques that are most commonly computed using the Jacobian transpose or similar abstractions [Wooten 1998; Faloutsos et al. 2001; Zordan and Hodgins 2002; Coros et al. 2010]. When motions are derived from motion capture data, the existing center-of-mass trajectory is commonly used as a reference and the parameters of the controllers can be optimized to yield the best overall behavior for a given motion or set of motions [Lee et al. 2010; Geijtenbeek et al. 2012].

Balance controllers can also be designed around *optimized inverse dynamics* methods, which either uniquely solve for, or optimize for, the joint torques at a given time step in order to achieve a combination of task goals and balance goals. This type of approach was first developed more than two decades ago [Stewart and Cremer 1992], and has since been extended in many ways that typically solve a quadratic program (QP) optimization at each time step [Abe et al. 2007; da Silva et al. 2008; Jain et al. 2009; Macchietto et al. 2009; de Lasa et al. 2010; Stephens and Atkeson 2010]. The control behaviors are authored indirectly through the design of the objective functions, and the regulation of the linear momentum is typically one of the key objectives. Because these methods optimize for the current time step, they cannot anticipate the upcoming motion except for the anticipation that is effectively designed into the objective function.

More recently, methods regulating *angular momentum* have been developed, which demonstrate impressive balance capabilities [Macchietto et al. 2009; Geijtenbeek et al. 2012; Borno et al. 2014]. Our work further enhances the capabilities of this control strategy by allowing for anticipatory behavior, as implemented via a motion-specific or task-specific optimization of the reference center-of-mass trajectory. This allows for balance during more dynamic motions and for motions that involve significant force-based interactions with the environment.

Model predictive control is another common methodology for balance control. This offers anticipatory control, as it involves solving for state trajectories and action trajectories for a forward-looking window of time. If this can be computed efficiently, the plan can be recomputed at every time step, i.e., online, in order to cope with unforeseen events as they arise. In the context of humanoid balance control, ZMP preview control [Kajita et al. 2003; Wieber 2006] is the most well known example and this has been applied with success in many humanoid robot controllers. The trajectories of the center of mass (CM) and center of pressure (CP) are both treated as free variables and are optimized to satisfy objectives such as minimal CP-excursion and minimal jerk. The simplest versions of the problem can be solved analytically for a point-mass model that remains at a constant height, while more complex versions are often posed as QPs, which often can be solved online. In another recent work an efficient algorithm to perform online trajectory optimization has been proposed to synthesize complex human behavior, such as getting up from an arbitrary pose on the ground [Tassa et al. 2012]. This method requires smooth differentiable evaluation functions.

In our work, we rely on a highly capable momentum balance controller to handle disturbances and opt to compute optimized desired CM trajectories offline using derivative-free methods. This methodology allows for more arbitrary constraints, objectives, and scenarios, including unknown contact-times and the non-linear dynamics of the full body that are not captured by the simplified models often used for preview control. These benefits come at the cost of requiring offline optimization to compute. However, we shall demonstrate that the the computed reference CM trajectories (CMRTs) can be generalized and interpolated to allow for online balance control of wider classes of motion.

Other approaches have also been explored, including controllers that have their design informed by biomechanics [Abdallah and Goswami 2005] or by motion capture data [Yin and Van De Panne 2006]. Animation also allows for kinematic-dynamic hybrid methods, where kinematic motions can be filtered [Tak et al. 2000; Yamane and Nakamura 2003; Tak and Ko 2005], sequenced [Zordan et al. 2005], modified [Metoyer et al. 2008], or optimized [Atkeson and Stephens 2007] to produce physically-plausible trajectories for balanced motions on a variety of models (some very much simplified, some not) and with varying degrees of generality. We focus our work on controllers that can be used with forward dynamics simulations, with the long term objective of being able to directly simulate wide classes of motion.

3 Momentum Balance Control

There are several features that are important for measuring the postural stability of a simulated character: the center of mass (CM), the center of pressure (CP), the ground reaction force (GRF), and the base of support (BS). A character has static postural stability if the gravity line from its CM falls within the BS defined by the convex hull of the feet, and the GRF at the CP in the BS passes through the CM [Shih 1996]. For a dynamic character, a criteria for stability is that its current state is within the set of viable states, i.e., those states for which a control trajectory exists that will prevent a fall [Wieber 2000]. For any given character with joint and torque limits, it is difficult to indentify the complete set of viable states and appropriate control trajectories. Instead, it is easier to design a controller that drives the character towards a configuration of static stability, even though this will only prevent falls on a subset of the viable states.

We use momentum control for character balance as introduced by Macchietto et al. [2009]. This controller moves the character toward static postural stability with a combination of linear and angular momentum control. Linear momentum control guides the center of mass on a trajectory defined by the differential equation

$$\dot{L}_{des} = k_l m(c_r - c) + d_l(m\dot{c}_r - L) \tag{1}$$

where m is the character mass, c_r and c are the reference and current CM positions, and k_l and k_d are gains that specify the temporal properties of the desired CM trajectory. The reference c_r is fixed with zero velocity at a position above the center of support, while the gains are designed to produce a slightly under-damped oscillation that quickly brings the CM to rest at the reference position. Angular momentum control is achieved by choosing a trajectory for the center of pressure which will keep it away from the edges of the BS, thereby avoiding situations where support rotation can occur.

An angular momentum change $\dot{H}$ will occur when external forces produce a torque about the center of mass. As such, the angular and linear controllers are coupled because the linear momentum control effectively dictates the GRF, and thus, through selecting the angular momentum $\dot{H}_{des}$, the ground force can be placed at a desired CP, which is likewise on a designed trajectory toward static stability.

Momentum control for balance also takes into account a reference character trajectory $\theta_r(t)$, though the motion may ultimately conflict with the goals of static and dynamic stability. The tracking control equation

$$\ddot{\theta}_{des} = k_t(\theta_r - \theta) + d_t(\dot{\theta}_r - \dot{\theta}) + \ddot{\theta} \tag{2}$$

can have low gains k_t and d_t due to the feedforward velocity and acceleration terms, thus allowing compliant reactions to perturbations. Macchietto et al. define three control objectives,

$$C_l = \|\dot{L}_{des} - \dot{L}\|^2 \tag{3}$$

$$C_h = \|\dot{H}_{des} - \dot{H}\|^2 \tag{4}$$

$$C_t = \|W(\ddot{\theta}_{des} - \ddot{\theta})\|^2, \tag{5}$$

where each is a quadratic function of the acceleration $\ddot{\theta}$ and dependent on the current state. They solve for the optimal acceleration $\theta^\star$ in the quadratic program

$$\min_{\ddot{\theta}} \beta_l C_l + \beta_h C_h + \beta_t C_t \quad \text{s.t.} \quad J_s \ddot{\theta} + \dot{J}_s \dot{\theta} = a \tag{6}$$

where the linear equality constraint uses the support-foot Jacobian J_s to ensure that the foot remains on a moving support surface with acceleration a. The weights β allow a tradeoff between tracking and balance, and the diagonal matrix W allows the tracking of some joints to be sacrificed more than others to minimize the balance objectives.

The optimal accelerations are typically computed at a lower rate than the simulation, which saves on computation and can introduces a natural delayed response to perturbation. However, the balance control is exclusively reactionary, and does not address the fact that it is sometimes advantageous to let the CM move away from the BS temporarily in order to improve stability, for instance, where there are interactions with the environment.

4 Reference Trajectory Synthesis

It is by modifying the CM reference trajectory in Equation 1 that we can give our balance controller the ability to anticipate. With anticipatory balance control, the character can throw its CM forward or backward in advance of interactions with the environment, or tricky balancing scenarios due to fast motions or unstable poses in the reference pose trajectory $\theta_r(t)$.

Given that the support can slip, we define the CMRT as a displacement relative to a coordinate frame at the center of the base of support. We use a smooth function to specify the trajectory, using a piece-wise cubic interpolating spline with varying knots. We typically use 4 control points because this provides the CMRTs a sufficient opportunity to pull back and forth on the CM several times (for instance, both in anticipation and in reaction); the small number of control points is also helpful in simplifying our search for optimal CMRTs as we discuss below. We constrain the spline curve to go through zero displacement with zero slope at the start and end because we only wish to introduce a temporary modification into the balance controller. Finally, we make the simplification that the trajectory only includes horizontal displacement. This is reasonable, noting that Macchietto et al. only track the CM projection in the ground plane. Likewise, it is horizontal displacements that have the largest direct effect on postural stability. Just the same, we note that vertical fluctuations of the CM could be important as they would generate fluctuations in the GRF that might be useful in some scenarios, such as to ensure the necessary friction forces for tracking motions with large angular components.

4.1 Balance performance optimization

With our strategy of modifying the balance controller, we still need a metric for determining what modifications are appropriate. We choose to measure the performance of the balance controller based on how far the CP strays from the center of the support. One interpretation of this is that the balance controller will be more robust in

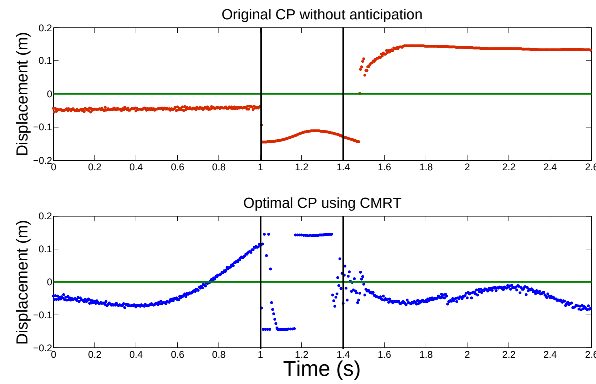

Figure 2: *Example CP trajectory optimization for the fast bending motion. The top and the bottom plots show the CP before and after applying CMRT.*

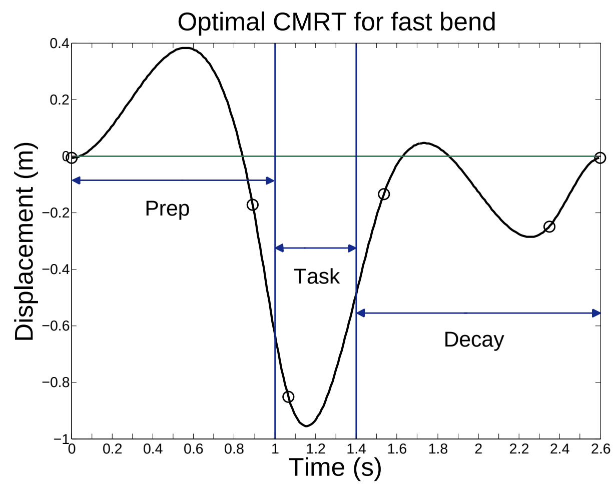

Figure 3: *Example CMRT trajectory for the fast bending motion.*

dealing with perturbations when the simulated character has its CP at the center of the base of support. That is, when the CP is near to the edge of the boundary of the support, it becomes easier for an external push to result in the undesired rotation of the support foot.

Given the CP trajectory $p(t)$ and the trajectory of the center of the support $s(t)$, we define a CP error trajectory $e(t) = p(t) - s(t)$. An example of the CP trajectory before the optimization run is shown in Figure 2 (top). For an input motion, such as a change of pose over a given duration, we create a simulation clip with a time interval of interest which includes a preparation time before the motion for anticipatory effects and a decay time after the motion to allow the character to approach static postural stability. We always use a preparation time of 1 second, while we use different decay times, typically 1 second or slightly longer depending on the duration of transients generated by the input motion. We run the simulation at a fixed step size to collect a time-discretized version of the clip's error trajectory, and then compute our performance measure as the 2-norm of the discretized error. This serves as our objective function in the optimization of CMRTs. Note that when no ground foot contact exists, we do not have a measurement of the CP error. While we could treat this as a type of failure, we instead remove these time steps from the error vector because we occasionally observe intermittent contact in our simulation. This is because we use the Open Dynamics Engine and choose to keep ground contact constraints firm with a very small constraint force mixing parameter of 7×10^{-5}.

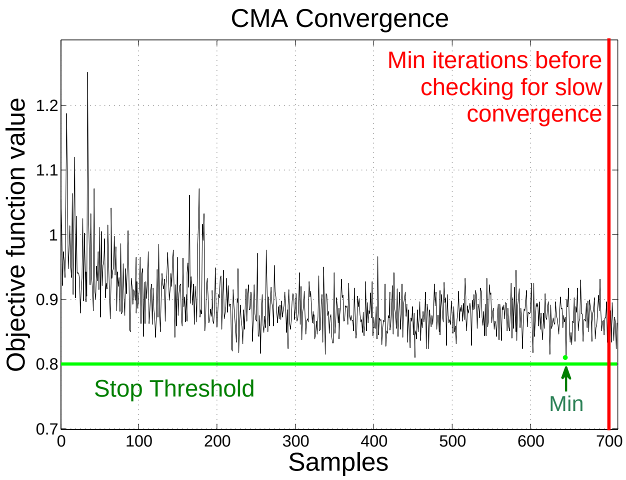

Figure 4: *A plot showing objective function values for all samples computed during a CMA optimization, along with the stop threshold value. In this case the minimum of 700 samples was passed, after which a test for slow convergence terminated the optimization.*

Given that our objective function involves a physics simulation of several seconds, we use the popular derivative free optimization method of Covariant Matrix Adaptation (CMA) [Hansen 2006]. Our parameter space has 8 dimensions consisting of the four control points and their knot values. We typically initialize the optimization with a flat spline with equally spaced knot values, but we have also used previous optimization results as starting points when computing CMRTs for pose changes at different speeds. As shown in Figure 3, CMRT has 3 phases, namely preparation, task and decay. Preparation is where the anticipation happens; the task refers to the motion of interest, either a pose change or an environmental interaction; and decay is where the character transitions between the anticipatory and the reactionary controllers. Figure 2 (bottom) shows the optimal CP error trajectory of the fast bending motion using CMRT.

Figure 4 shows convergence of the optimization for a typical run. We use a stopping threshold to permit an early optimization exit for a result known to be good. We otherwise check for slow convergence during optimization, but only after a minimum number of samples have been computed across all of its iterations (typically 700). The stopping threshold depends on dimension of the discretized error trajectory (i.e., the clip length and step size) and we set the value based on experience.

The run-times of the optimizations given in Table 1. Motions typically require 1-10 minutes to optimize. However, the compute time for the bending motions demonstrates one of the more difficult cases. The optimization times are large because the input motion is harder to balance. The interpolated motion of the start and end poses has the ground-plane projected CM traveling mostly outside the base of support. Because the motion is difficult to balance, fewer of the CMRTs randomly generated during the optimization process will result in good balance behaviors.

4.2 Interpolation and Superposition of CMRTs

Figure 5 shows several CMRTs optimized with CMA for a bending pose change at different speeds. Note how the anticipation changes across the different speeds, with the first knot occurring later for the slower motion. Furthermore, the CMRT curves have smaller amplitudes for the slower motions.

input motion	L	P	T	D	δ	N	C
push dominoes	3.0	1	0.35	1.65	0.6	149	328
catch ball	3.0	1	0.3	1.7	0.9	18	40
slow bend	2.6	1	1.2	0.4	0.9	328	641
fast bend	2.6	1	0.4	1.2	1.1	683	1366

Table 1: *CMRT computation times for different motion clips, where L is the clip duration (including prep, task and decay times), P is the preparation time, T is the task duration (task or pose change), D is the decay time, δ is the CMA convergence threshold, N is the number of iterations computed, C is the computation time, and all times are in seconds.*

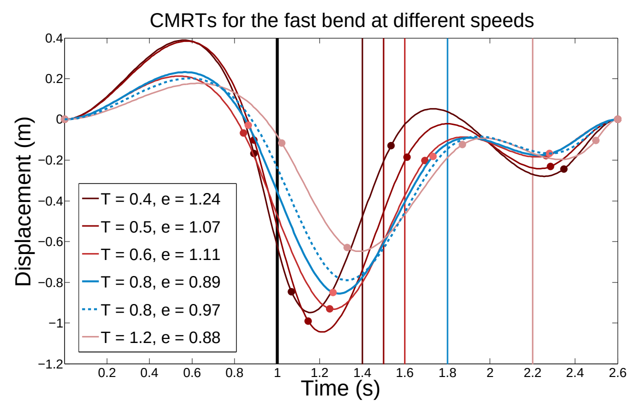

Figure 5: *Example CMRT trajectories for different speeds of the same motion clip. The start of the task motion is the same for all and happens at 1 s, shown by a solid black line. The end of the task motion is different for different motion speeds and are shown by the vertical solid lines. The legend specifies the task speed in seconds (T), and the objective function value (e). The interpolated CMRT is shown in blue dashed line and is compared to the optimization solution shown in blue solid line.*

In order to produce a CMRT for a pose interpolation at an intermediate speed, we interpolate the optimal solutions found at adjacent speeds. We perform the interpolation using the knots values and control points. This allows the knot values to shift to accommodate a varying anticipatory control at different speeds. Figure 5 also shows an example of the interpolated curve in dashed lines. The legend of the figure demonstrates that the objective function value for interpolated and optimized curves are quite similar.

Up to now we have only explored how to modify balance control for individual input motions, but in a real usage scenario we would like to combine many input motions into interesting sequences. Our solution for this is to superpose the displacements of adjacent CMRTs when they overlap. This is a plausible strategy under the assumption that it allows the anticipation of the next control modification to add to those happening in the decay time of the previous. We expect that one of the reasons why this can work well is that the optimal CMRTs in the decay window tend to be small. Figure 6 shows an example of superposed CMRTs of 4 changing pose motions.

5 Results

We present results for a variety of motions and tasks, which are best seen in the supplementary video.[1] We first demonstrate that

[1] http://www.cs.mcgill.ca/~kry/pubs/abc/

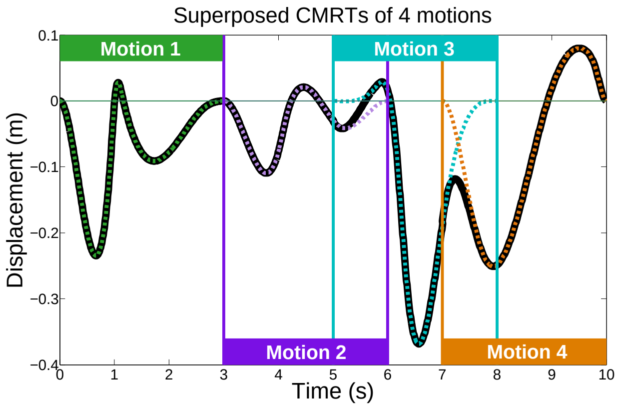

Figure 6: *Superposed CMRTs of 4 motions.*

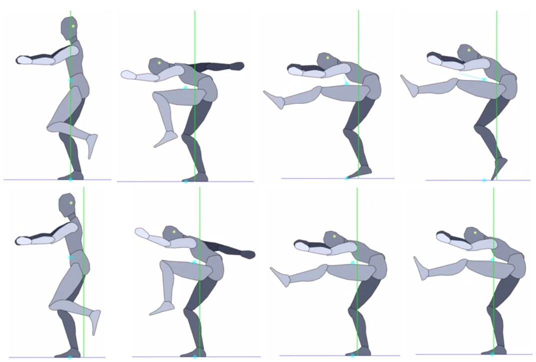

Figure 7: *Fast pose-to-pose transition. Top: Without anticipation. Bottom: With anticipation.*

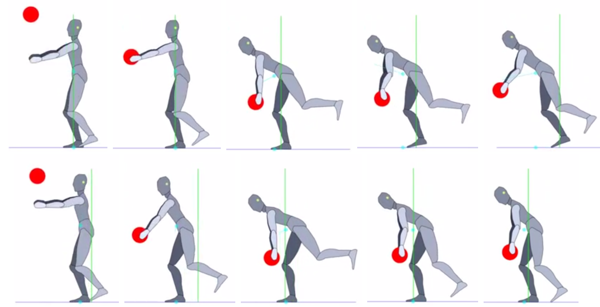

Figure 8: *Catching a heavy ball. Top: Without anticipation. Bottom: With anticipations.*

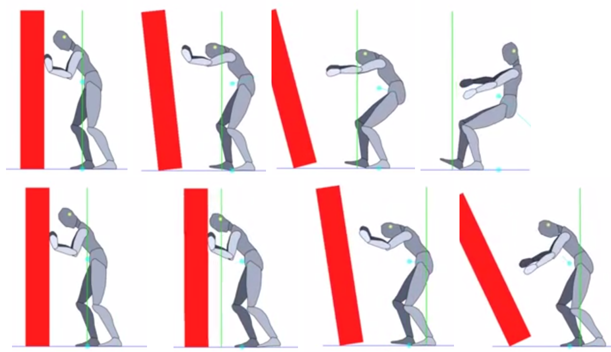

Figure 9: *Pushing a heavy object. Top: Without anticipation. Bottom: With anticipation.*

we can reproduce results similar to those demonstrated by Macchietto [2009]. The desired kinematic trajectories for the motions are authored using ease-in-ease-out key frame interpolation. The CMA objectives are as outlined earlier, with the exception of the ball catching example, where the objective heavily penalizes solutions when it is not feasible to catch the ball.

Dynamic motions: We author a number of highly dynamic motions using pose-to-pose transitions. One of these is shown in Figures 7. In this figure, the two cyan dots mark the center of mass location and its ground-plane projection, while the green vertical line illustrates the center-of-mass reference location. The anticipation via the CMRT is critical to the success of the motions. Executing the desired pose change without anticipation results in a loss of balance in the forward direction because of the fast leg swing. The CMRT anticipates this and compensates by moving the center of mass to the right in the early phases of the motion. Pose transitions can be concatenated together in sequence, with the character briefly coming to rest at each desired pose, as shown in Figure 6 and in the supplementary video.

Interpolation: The spline knots that define the CMRTs can be interpolated to yield valid and effective CMRTs for intermediate motions. We evaluate this using pose-to-pose transitions that are linearly retimed to produce a family of motions at different speeds. Figure 5 compares the fidelity of the interpolated CMRT with CMRT that is specifically optimized for the given motion speed.

Heavy ball catch: Figure 8 shows the effect of anticipation when catching a 7 kg ball. The weight results in a forward loss of balance that is not within the capabilities of the momentum-based controller if it is seen simply as a disturbance. The CMRT draws the center of mass to the right in anticipation of the impact of the catch. If the ball ends up being much lighter than expected, this compensation then becomes problematic and the character falls backwards as a result.

Domino push: This example involves a push that topples a heavy 140 kg object. Pushing without anticipation causes the character to fall backwards as a result of the push. The optimized reference trajectory shifts the CM forwards early on in the motion, thereby helping to use the character's weight during the push, as well as allowing the action of the push to result in the restoration of balance.

6 Conclusion

We introduce and evaluate a method that enables balance controllers to be anticipatory. For a given desired motion or task, the center-of-mass reference trajectory is optimized to achieve the best performance. This allows for more dynamic motions than can be achieved with non-anticipatory balance models, and also allows for standing tasks that can exploit center-of-mass manipulation in order to perform better at force-based interactions with the environment. Our work can also be seen as introducing an offline version of "full body" preview control to the control of balance for physics-based animation, coupled with a highly robust angular-and-linear momentum balance strategy. This allows for classes of motion that would otherwise fail with online preview control methods that use simplified models, such as a fast rotation to a crouched position, dynamic pushes, and more. The anticipatory shifts of the center of mass further reflect a simulated character's intent and expectations in performing a given action. Lastly, we demonstrate that the learned reference trajectories can be successfully generalized using interpolation and extrapolation.

There exist many directions for future work. We wish to extend the method and results to fully 3D examples. Our anticipatory balance control can also be integrated with balance models that take one or more steps in order to recover balance, and systems that are equipped with a collection of other controllers. Currently the method assumes that one or both feet are in contact with the ground. We wish to explore extensions of the method that allow for flight phases and other forms of intermittent contact with the ground or the environment. It is likely possible to develop online versions of motion-or-task-specific anticipatory balance control with appropriate task abstractions and dynamic abstractions. We also wish to better understand the sensitivity of human observers to perceiving shifts of the center of mass in various contexts, and to understand how anticipatory balance control is achieved in human motor behaviors.

7 Acknowledgments

We thank the anonymous reviewers for the suggestions for improving the paper. This work was supported by funding from NSERC and GRAND NCE.

References

ABDALLAH, M., AND GOSWAMI, A. 2005. A biomechanically motivated two-phase strategy for biped upright balance control. In *IEEE Intern. Conf. on Robo. and Auto.*, 1996–2001.

ABE, Y., DA SILVA, M., AND POPOVIĆ, J. 2007. Multiobjective control with frictional contacts. In *SCA '07: Proc. of the 2007 Symp. on Comp. Anim.*, 249–258.

ATKESON, C. G., AND STEPHENS, B. J. 2007. Multiple balance strategies from one optimization criterion. In *Humanoids*, 57–64.

BORNO, M. A., FIUME, E., HERTZMANN, A., AND DE LASA, M. 2014. Feedback control for rotational movements in feature space. *Comp. Graph. Forum 33*, 2, 225–233.

COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2010. Generalized biped walking control. *ACM Trans. on Graph. (TOG) 29*, 4, 130:1–130:9.

DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Simulation of human motion data using short-horizon model-predictive control. *Comp. Graph. Forum 27*, 2, 371–380.

DE LASA, M., MORDATCH, I., AND HERTZMANN, A. 2010. Feature-based locomotion controllers. *ACM Trans. on Graph. (TOG) 29*, 4, 131:1–131:10.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *SIGGRAPH '01: Proc. of Comp. Graph. and Inter. Tech.*, 251–260.

GEIJTENBEEK, T., PRONOST, N., AND VAN DER STAPPEN, A. F. 2012. Simple data-driven control for simulated bipeds. In *Proc. of the ACM Symp. on Comp. Anim.*, SCA '12, 211–219.

HANSEN, N. 2006. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, vol. 192. Springer Berlin / Heidelberg, 75–102.

JAIN, S., YE, Y., AND LIU, C. K. 2009. Optimization-based interactive motion synthesis. *ACM Trans. on Graph. (TOG) 28*, 1, 10:1–10:12.

KAJITA, S., KANEHIRO, F., KANEKO, K., FUJIWARA, K., HARADA, K., YOKOI, K., AND HIRUKAWA, H. 2003. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE Intern. Conf. on Robo. and Auto.*, vol. 2, IEEE, 1620–1626.

LEE, Y., KIM, S., AND LEE, J. 2010. Data-driven biped control. In *ACM Trans. on Graph. (TOG)*, 129:1–129:8.

MACCHIETTO, A., ZORDAN, V., AND SHELTON, C. R. 2009. Momentum control for balance. *ACM Trans. on Graph. (TOG) 28*, 3, 80:1–80:8.

METOYER, R., ZORDAN, V., HERMENS, B., WU, C.-C., AND SORIANO, M. 2008. Psychologically inspired anticipation and dynamic response for impacts to the head and upper body. *IEEE Trans. on Vis. and Comp. Graph. 14*, 1, 173–185.

SHIH, C.-L. 1996. Analysis of the dynamics of a biped robot with seven degrees of freedom. In *IEEE International Conf.on Robo. and Auto.*, 3008–3013.

STEPHENS, B. J., AND ATKESON, C. G. 2010. Dynamic balance force control for compliant humanoid robots. In *IEEE Intern. Conf. on Intel. Robo. and Sys.*, IEEE, 1248–1255.

STEWART, A. J., AND CREMER, J. F. 1992. Beyond keyframing: An algorithmic approach to animation. In *Proc. of Graph. Inter.*, 273–281.

TAK, S., AND KO, H.-S. 2005. A physically-based motion retargeting filter. *ACM Trans. on Graph. (TOG) 24*, 1, 98–117.

TAK, S., SONG, O.-Y., AND KO, H.-S. 2000. Motion balance filtering. *Eurographics 19*, 3, 437–446.

TASSA, Y., EREZ, T., AND TODOROV, E. 2012. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IROS'12*, 4906–4913.

WIEBER, P.-B. 2000. Constrained dynamics and parametrized control in biped walking. In *Intern. Symp. on Math. Theo. of Net. and Sys.*

WIEBER, P.-B. 2006. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *IEEE Intern. Conf. on Hum. Robo.*, IEEE, 137–142.

WOOTEN, W. 1998. *Simulation of Leaping, Tumbling, Landing, and Balancing Humans*. PhD thesis, Georg. Tech.

YAMANE, K., AND NAKAMURA, Y. 2003. Dynamics filter - concept and implementation of on-line motion generator for human figures. *IEEE Trans. on Robo. and Autom. 19*, 3, 421–432.

YIN, K., AND VAN DE PANNE, M. 2006. Omnidirectional humanoid balance control: Multiple strategies for reacting to a push. Tech. Rep. TR-2006-11, University of British Columbia.

ZORDAN, V. B., AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *ACM Symp. on Comp. Anim.*, 89–96.

ZORDAN, V. B., MAJKOWSKA, A., CHIU, B., AND FAST, M. 2005. Dynamic response for motion capture animation. *ACM Trans. on Graph. (TOG) 24*, 3, 697–701.