Dynamic Humanoid Balance through Inertial Control

Odest Chadwicke Jenkins #1, Pawel Wrotek #2, Morgan McGuire *3

#*Department of Computer Science, Brown University*
Providence, RI, USA 02912-1910
[1]cjenkins@cs.brown.edu
[2]pwrotek@gmail.com

**Computer Science Department, Williams College*
Williamstown, MA, USA 01267
[3]morgan@cs.williams.edu

***Abstract*— Physical humanoids often require the ability to maintain upright balance while performing various tasks involving locomotion and environmental interaction. Such balance requirements have been difficult to maintain with traditional approaches to articulated motion control. We claim that these difficulties are significantly due the use of parent-space in standard proportional-derivative (PD) servoing, typically requiring highly sophisticated decision making policies to function while maintaining balance. Using inspiration from inverted pendulum robots, we address humanoid balance control through a world-space servoing model. Our model retains the same basic form as the PD-servo, but uses inertial/accelerometer measurements rather potentiometer-like sensing. Our humanoids are able to functionally balance, locomote, and recover without sophisticated decision making. We demonstrate the efficacy of our approach through simulation experiments involving locomotion, user interaction, ballistic motion, uneven terrain, and dramatic disturbances.**

Fig. 1. Screenshot from two physically-simulated inertially-controlled boxers fight on an uneven platform.

I. Introduction

To a simple observer, it seems intuitive that a humanoid robot with infinitely powerful motors and no joint angle limitations should be able to move around its environment by its own power. Further, such robots should be able to perform physically superhuman feats beyond basic humanoid actions, such as walking, jumping, and manipulating other objects. While infinitely strong robots do not exist in reality, physically simulated humanoids are common in virtual environments. These humanoids are becoming increasingly prevalent due to the rise of physical simulation in video games, computer animation, and robotics research.

For simulated humanoids, we claim motion control is more of an issue for servo mechanisms rather than high-level decision making and planning. Despite having unincumbered actuation, however, current virtual humanoids struggle to perform the simplest of activities. In particular, humanoids have difficulty in executing desired motion trajectories while maintaining upright balance. We attribute this problem to the standard use of the proportional-derivative (PD) servo for generating motor forces from desired poses. About each degree-of-freedom (DOF), the standard PD-servo minimizes *parent-space error*, the difference between desired and actual angles expressed in the coordinates of the parent bone. Parent-space error incorporates no knowledge about *extrinsic* pose (relation to global coordinates) and only accounts for *intrinsic* pose (joint angle orientation). Consequently, extrinsic constraints such as balance must be accounted for by more sophisticated decision making. Additionally, the PD-servo is essentially a spring-and-damper model with gain coefficients that can be difficult to set properly.

In this paper, we present a *world-space* approach to humanoid motion control capable of balance, locomotion, and emergent environmental interaction without sophisticated decision making. Our world-space servo retains the same basic form as the PD-servo, but defines error with respect to global coordinates. We analogize world space error to the sensing the orientation of each bone with inertial sensing (accelerometers), as opposed to potentiometers for sensing joint angles. Through our inertial control, maintaining extrinsic pose constraints, such as upright balance, is implicit in the feedback error. Given no motor or pose restrictions, inertial control provides a simple servoing mechanism suitable for dynamic balancing in a variety of

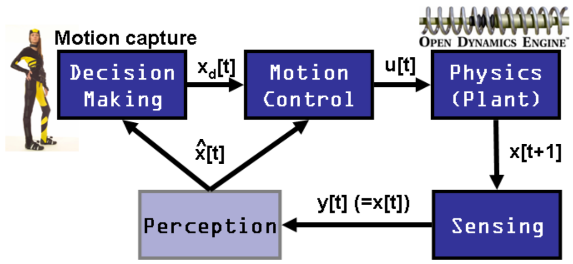

Fig. 2. An overview of our approach with a feedback control loop. Motion capture provides trajectories that are controlled through world space servoing and physically integrated by the Open Dynamics Engine. We assume the humanoid has full observability.

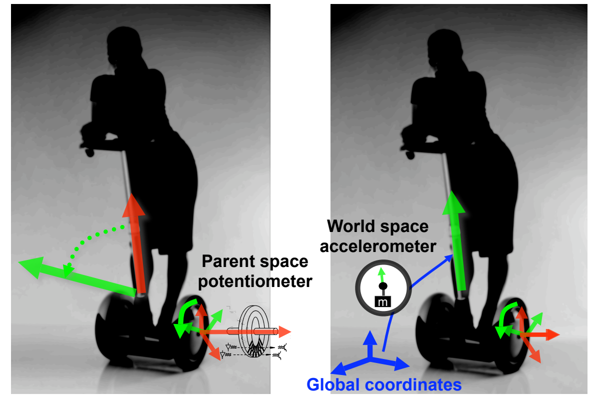

Fig. 3. Our inspiration: inverted pendulum control. Parent space control is similar to servoing based on the orientation of the rider with respect to the wheel. This does not consider whether the rider remains upright and must be accounted for by a decision making system. World space control senses the rider's global orientation with respect to gravity and, thus, simple inertial feedback error can keep the rider upright.

locomotive tasks. Further, inertial servoing is less sensitive to the selection of motor gains than parent space approaches. We demonstrate the efficacy of inertial servoing through simulation experiments involving locomotion, emergent behavior with new objects, user interaction, ballistic motion, uneven terrain, and dramatic disturbances. These experiments use motion capture as a open-loop decision making policy.

II. Background

The problem we address is motion control of an autonomous infinitely strong humanoid. Shown in Figure 2, motion control is one component in a (closed-loop) feedback control loop. The purpose of motion control is to generate a set of motor forces $u[t]$ at time t to move towards a desired configuration $x_d[t]$ given an observed configuration $\hat{x}[t]$. Desired configuration $x_d[t]$ and observed configuration $\hat{x}[t]$ are given by a (typically goal-driven) decision making policy and a perception procedure, respectively. Motor forces $u[t]$ and current system state $x[t]$ are integrated with respect to physics over time to produce the updated system state $x[t+1]$. For this work, we assume system state is fully observable ($\hat{x}[t] = y[t] = x[t]$), desired configurations are governed in an open-loop manner by motion capture data, and a physics engine for simulating rigid-body dynamics.

Control of physically simulated humanoids was pioneered by Hodgins et al. [1]. This work presented manually crafted procedures to control humanoids performing involving dynamic balance, such as running and cycling. Their controllers consisted of finite-state machine decision policies that set desireds executed by parent-space PD-servo motion controllers. Dynamic balance arises directly from the manual efforts of the implementer. Zordan and Hodgins [2] later automated several aspects of this approach, including automated selection of motion gains, the use of motion capture in finite states, and static balance using virtual actuator control [3].

Several approaches to dynamic balance cast the problem as an issue for decision making, where a plan of actions is determined. Zero moment point (ZMP) control [4] is the most widely utilized of these approaches for both real and virtual humanoids. ZMP control plans motion trajectories such that humanoid's center of pressure remains over its polygon of support with its contact surface, assuming complete knowledge of the robot's dynamics. Hofmann and Williams [5] plan "flow tubes" based on the concepts of funnel-style attractors [6] and related notions of a control basis [7]. Similarly, Faloutsos et al. [8] use a Support Vector Machine to find valid switching decision making system for transition between existing attractors.

Alternatively, other approaches attempt to learn or develop adaptive dynamic balancing policies. Tedrake et al. [9] use reinforcement learning made tractable by inspirations from the mechancal redundancy of passive walkers. Ramamoorthy and Kuipers [10] define families of harmonic primitives for passive walkers on uneven terrain. Endo et al. [11] learn central pattern generators with a policy gradient system. Given full knowledge of the robot's inverse dynamics, which could be learned by [12], control forces to perform dynamic balance can be explicitly computed or optimized. However, analytical inverse dynamics can be complicated to formulate and learned models are specific to only behavior previously observed.

III. Inverted Pendulum Analogy

Inertial motion control is inspired by how inverted pendulums (e.g. a Segway HT) dynamically balance and generalizes this notion to humanoids. Illustrated in Figure 3, an inverted pendulum robot attempts to keep its rider upright by applying appropriate motor torques to its powered wheels, which grip to the ground surface. The pendulum's motion control mechanism is continually computing torques to correct the rider's orientation with respect to the ground towards being upright (perpendicular to the ground). Traditional approaches would use a parent-space servo, minimizing angular error between the wheel (parent) and the rider (child) as sensed by a potentiometer on the wheel axis. In the naive case, the wheel's coordinates will rotate when the wheel spins. This will cause the rider's desired orientation to rotate in world coordinates and quick loss of upright balance. The problem is that motion controller has no sense of what direction is "up". This lack of "up" could be addressed using an parent-space sensor that remained fixed with respect the ground, such as an appropriately mounted optical encoder. However, this relationship to "up" is only due the wheel's direct contact with the ground and would not scale to sensing "up" for every part in a humanoid.

In reality, inverted pendulums use inertial sensing (e.g., accelerometers) to sense their global orientation. Accelerometers sense their orientation using the Earth's gravitational field. Thus, gravity provides the world-space reference frame for upright motion control, similar to the vestibular system in the human inner ear. This begs the question: "What would happen if motion control frame its error in inertial coordinates?" Inertial motion sensing is readily accessible with current commercial products. We hypothesize that given infinite strength and no joint limits, a robot could instead use inertial sensing directly for motion control and implicitly perform dynamic balance.

IV. Inertial Motion Control

A standard proportional-derivative (PD) motion controller computes motor torques about each degree of freedom based on the desired and actual angles and angular velocities:

$$\text{torque} = k_1(\theta_{\text{desired}} - \theta_{\text{actual}}) + k_2(\dot{\theta}_{\text{desired}} - \dot{\theta}_{\text{actual}}) \quad (1)$$

k_1 and k_2 are user-specified motor gains with respect to position and velocity Variable and θ is only for notational clarity. Equation 1 is directly applicable to parent-space hinge joints, which are parameterized by a single joint angle that is easily measured. Like previous work [2], we use PD controllers, but must generalize from 1 DOF hinge joints to 3 DOF ball joints For a given simulation step, we denote the known quantities *at each bone* as: $\mathbf{P}_d$ desired *parent-space* orientation, $\mathbf{W}_d$ and $\mathbf{W}_a$ desired and actual *world-space* orientation, and ω_d and ω_a world-space angular velocities. Our matrix multiplications assume column-vector matrices, where a vertex to be transformed would appear on the right of the matrix. From these variables, we solve for world-space torque τ. Unless otherwise specified, these variables refer to the current bone b; alternately an additional subscript specifies the parent or root bone. Operating on 3D orientation matrices instead of 1D angles, it is necessary to express the differences in equation 1 as a function mapping two matrices to a 3-vector whose direction $\vec{v}$ is the axis of rotation between the reference frames and whose magnitude is the rotation angle θ. We denote the difference: $\Delta(\mathbf{D},\mathbf{A}) = \theta\vec{v}$, where θ and $\vec{v}$ are the axis and angle of the product $\mathbf{D} * \mathbf{A}^{-1}$.

Humanoids apply torques to their bones to match target motion capture or key frame poses. Motion capture data typically is expressed as a series of key poses, containing a coordinate frame for each bone relative to its parent. Poses between key poses are obtained by spline interpolation. Each bone's target orientation frame is lifted from the parent space matrix $\mathbf{P}_d$ to world space matrix $\mathbf{W}_d$ by recursively applying forward kinematics. The *actual* orientation of all bones except the root are ignored by this transformation. This is the key to the stability for inertial control. Previous methods compute a target relative to the current parent reference frame, which propagates error down long linkages.

We define the desired world-space angular velocity ϖ as the angular velocity needed to reach the desired pose at some future time $(t+\delta)$ given the current desired pose,

$$\omega_d = \frac{1}{\delta}\Delta\big(W_d(t+\delta), W_d(t)\big). \quad (2)$$

In our simulations, we chose δ to be the duration between mocap keyframes. Note that we did not choose the instantaneous derivative of the current desired orientation as desired velocity. Because physics integrates state in discrete Euler time steps, it is necessary to look at least one timestep into the future for velocity, since that is when the effects of the applied torque will be observed.

The world-space torque τ to apply to the bone servo for the next simulation step is given by,

$$\tau_{b\neq\text{root}} = k_1\Delta(\mathbf{W}_d, \mathbf{W}_a) + k_2(\omega_d - \omega_a). \quad (3)$$

This is a straightforward world-space, 3D variation on eq. 1. Gain constants used in our experiment are described in Section V. Because error is not propagated through the bone hierarchy, the constants need not be tuned precisely. Additional consideration should be given toward whether world space is defined absolute or egocentric coordinates. In absolute coordinates, the servo routine will enforce the character's pose such that the orientation of the root aligns with motion capture. Described in our other work (blind citation), world space control leads to issues with "super-balancing", which we address using an egocentric world space (or "person coordinates"). Motion control in person space leaves the root orientation free to be commanded by higher-level decision making.

V. Results

We implemented a test framework for inertial control in a physically simulated virtual environment. Rigid body physics are provided by the Open Dynamics Engine (ODE; http://ode.org). The G3D library (http://g3d-cpp.sf.net) is used for rendering and general 3D support code. Tests were run on a single-core 3.5 GHz Intel Pentium 4 processor under Windows XP. Humanoid kinematics and motion were provided through mocap data obtained from the CMU Motion Capture Database, Credo Interactive's MegaMocap V2 package, and a custom animated "stand up" motion. Figures in this section are excerpts of our comprehensive results video. Most show simple scenes for clarity; we can simulate significantly more complex cases.

Our world space controllers require no artificial help for maintaining static balance (i.e., when the center of mass is between the points of support). We compare them to parent-space control methods that **are** aided by a global "meathook" in

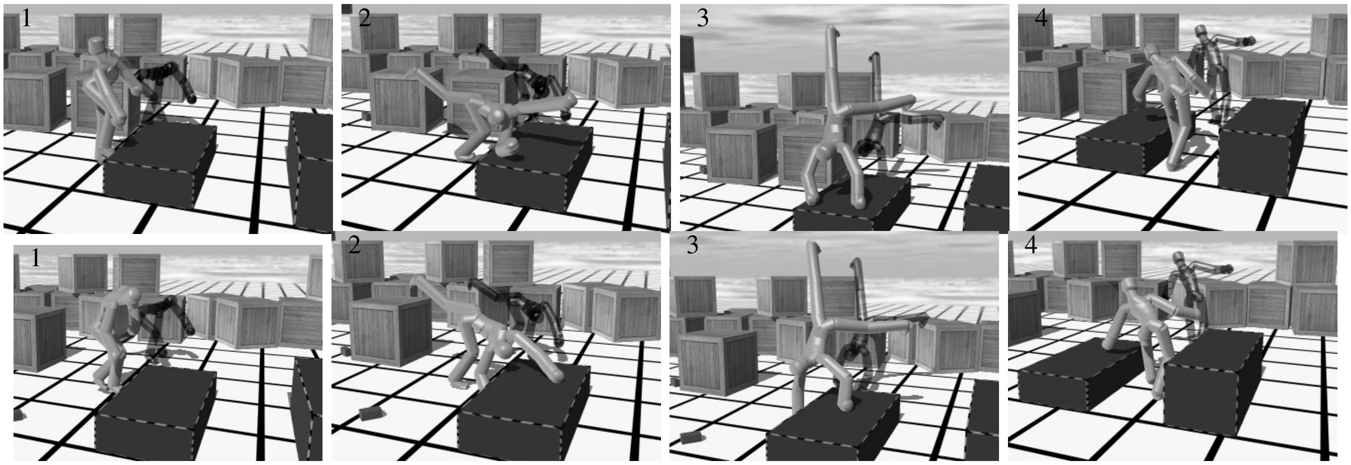

Fig. 4. Sequences of four frames showing parent-space (top) and world-space (bottom) humanoids adapting a cartwheel animation to an obstacle. The mocap driving the humanoids is shown behind in black. Both methods produce reasonable results, however, world-space method better preserves the style of the original motion.

order to provide a usable baseline; the parent space characters would otherwise fall over before any comparison could be made! For some dynamic balance situations a root spring applies torques at the root, however these are torque limited and could theoretically be propagated backwards to the feet in most cases. Doing so in cases where dynamic balance is physically achievable is future direction for our work.

Gain insensitivity for inertial control was evaluated with respect to cartwheel and obstacle navigation tasks involving dynamic balance. Shown left-to-right in Figure 4, humanoids are executing motion capture with no simulation, parent-space servoing, and world-space control. Significant effort was spent finding the best servo gain constants for critical parent-space damping. The world space method is sufficiently insensitive that we experimented with both custom gains as well as direct usage of the parent-space gains. Both sets of gains produced similar results for world space servoing. Figure 4 also illustrates emergent behavior for adapting to an obstacle. If the humanoid is located in a flat, featureless environment, world-space drives it to perform the cartwheels just as the original motion indicates. However, if there is an obstacle in the way our humanoid now reacts accordingly and realistically vaults over the obstacle. Even when balanced artificially, the parent-space adaptation loses much of the style from the mocap. We further explored this behavior in an obstacle course task (Figure 5), where world space servoing demonstrated a clear improvement to parent space. The humanoid for this task was required to walk up a ramp, down stairs, through a hanging crate, and finish with cartwheels over large blocks in the walking path. During its navigation of the course, the world-space humanoid remains true to the motion capture animation, but the parent-space humanoid has less stability and appears to quiver and jiggle.

Static balance for world-space control was tested subject to interactive applications of force. Figure 6 shows a world-space humanoid maintaining static balance despite strong external forces and user-imposed constraints. The humanoid was in an environment littered with obstacles and violently dragged about with a mouse-driven cursor. When pressed into the ground, the humanoid bends at the knees and then stands upright when released. During this event, the humanoid is ballistic for a short period and is able to properly recover.

Dynamic balance and ballistic motion for our humanoids as tested using motion and scenes involving locomotion and jumping. Figure 5 shows our humanoid using a flat ground walking motion to traverse uneven ground and walk through obstacles. Because the volume of the capture subject is not an exact match to the humanoid geometry, locomotion often incurs premature foot contacts with the ground. In place of smarter decision making or better foot geometry, we apply a small compensatory upward force on a foot with forward momentum that contacts the ground. Although not shown, we performed tests with an inertially-controlled humanoid driven by a jumping motion. The character achieves flight off of the ground and is able to stay upright upon landing. However, the character lags behind the mocap, which we attribute to overcoming gravity and the physical differences of the humanoid and the capture subject.

Recovering balance was evaluated through humanoid performance when violently impacted with various objects. Figure 7 shows three humanoids (world space, person space, and root-spring) repeatedly hit with "beach ball" objects and finally impacted with a crate. The root spring humanoid is artificially balanced and lowers its motor gains when "knocked out". Each root controller maintains balance during the light impacts of the beach balls. The heavier crate impact demonstrates the spectrum of reactions we can produce, varying from recovering and maintaining upright balance, falling with continuing actuation, and falling to unconsciousness.

Dynamic interactions over uneven terrain were evaluated

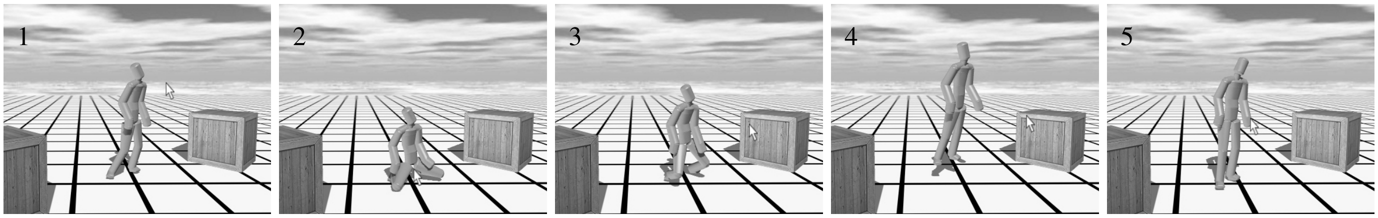

Fig. 6. Sequence of an world-space humanoid maintaining balance after being manipulated by a user controlled cursor.

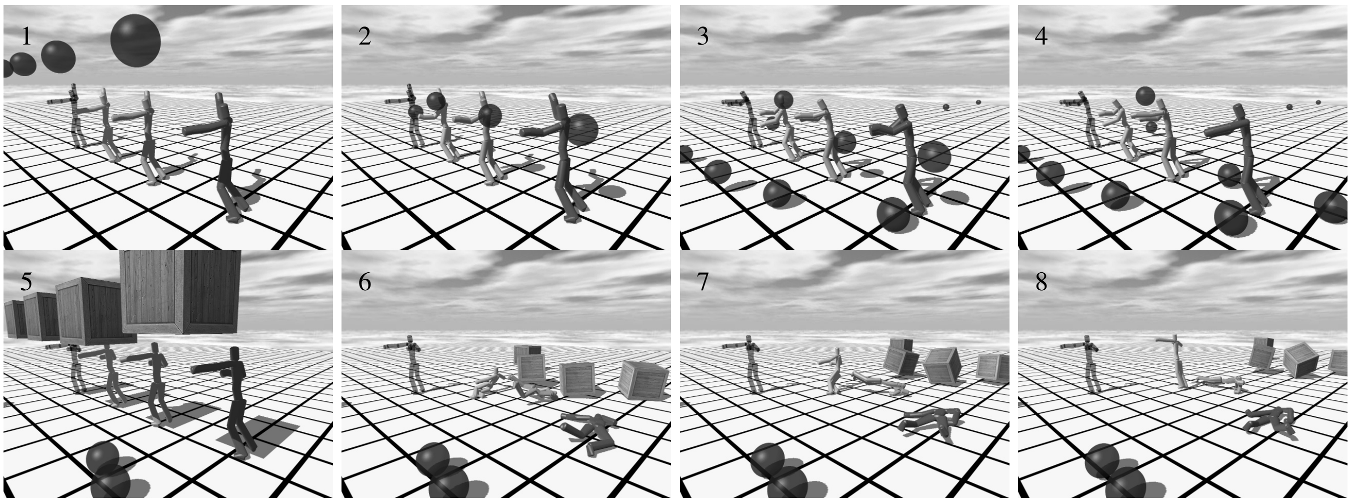

Fig. 7. Eight frames, spaced 0.1s apart, of four humanoids executing mocap while being hit with various objects. Characters use the following algorithms (left to right): mocap with no simulation, world space control, person space control, and a root-spring control that goes "unconscious" on heavy impacts. They are buffeted by the balls but only knocked over by the crate.

in an "boxing" scenario, where two autonomous humanoids "fought" against each other on an uneven set of crates. Each fighter used world space control and decision making based on random traversal of a motion graph [13]. The motion graph was constructed from mocap of an extended sequence of punches and a standing up motion. The boxers perform various punching motion until stunned and knocked down by a punch from the other character. Once down, the character takes desired motion from the standing up motion and returns to the punch part of the motion graph. Figure 8 shows a frame-by-frame comparison of world space boxers with straight mocap playback.

VI. Conclusion

In this paper, we have addressed the issue of coordinate spaces for low-level motion control, namely for dynamic balance. The parent-space formulation of the PD-servo is a holdover from rotational sensor limitations that no longer exist. As sensing and localization technologies improve and get smaller, motion control systems will have a more diverse set of modalities from which to generate motor behavior. We will not be simply restricted to control over individual joint angles or endeffectors through inverse kinematic. Instead, various modalities of motor behavior can be explored through setting feedback error different coordinate spaces. Utilizing advances in accelerometer technology, we have presented world space control as one alternative coordinate space that allows for emergent dynamic balancing. Another potential coordinate alternative could be teacher-space for performing imitation learning. Such coordinate space alternatives will offer greater synergy for humanoid decision making systems. As new control coordinates emerge, decision making systems will be able choose coordinates conducive to achieving specific goals, rather than creating complex plans in awkward control spaces. For inertial control, a clear synergy arises that allows decision making to focus on goal-oriented behavior in global coordinates. Decision making takes as input the current state of the world and output the desired global coordinate frame and bodycentric pose, which can then be executed by inertial control.

References

[1] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien, "Animating human athletics," in *Proc. SIGGRAPH '95*. New York, NY, USA: ACM Press, 1995, pp. 71–78.

[2] V. B. Zordan and J. K. Hodgins, "Motion capture-driven simulations that hit and react," in *ACM SIGGRAPH Symposium on Computer Animation*, 2002.

[3] J. E. Pratt, "Virtual model control of a biped walking robot," Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1995.

[4] M. Vukobratovic and B. Borovac, "Zero-moment point-thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

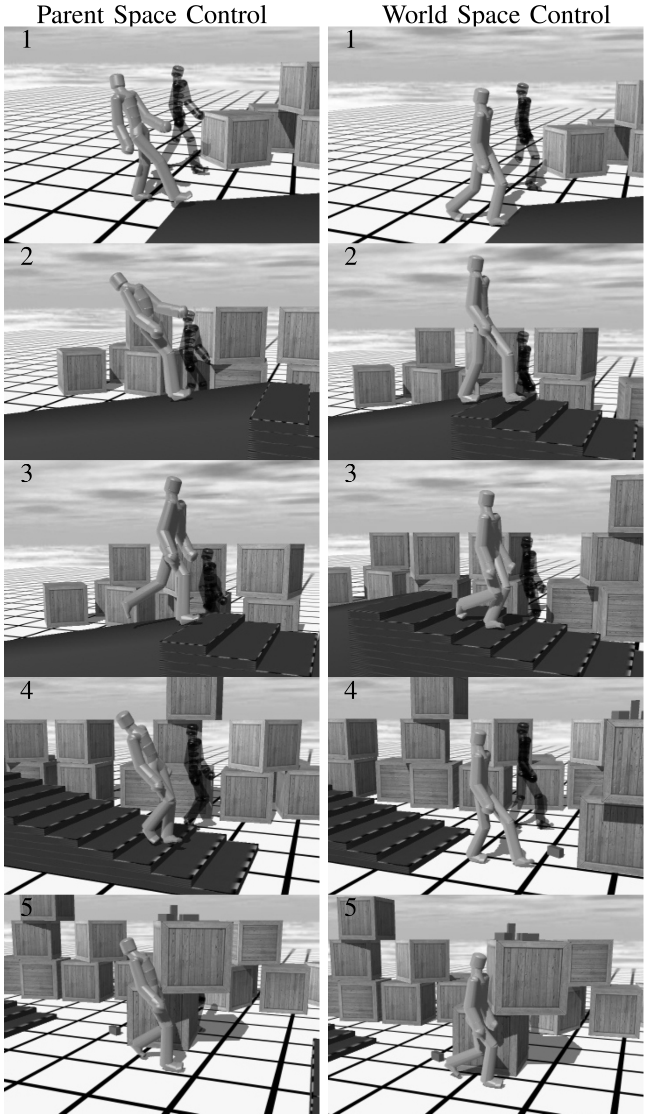

Fig. 5. Two sequences of walking an obstacle course, comparing parent space (left) and world space (right) control. Mocap ghosts are shown in the background for comparison. World space servoing produced faithful motion across various damping parameters. Parent space incurred wobble in the torso, despite appropriate damping.

Fig. 8. Comparison of two boxing humanoids using mocap (left) and world space control (right). World space properly handles mocap artifacts such as implausible foot contacts, character inter-penetration, and overall lack of a reaction to hits.

[5] A. G. Hofmann and B. C. Williams, "Exploiting spatial and temporal flexibility for plan execution for hybrid, under-actuated robots." in *National Conerence on Artificial Intelligence*, 2006.

[6] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[7] M. Huber, W. MacDonald, and R. Grupen, "A control basis for multilegged walking," 1996. [Online]. Available: citeseer.ist.psu.edu/huber96control.html

[8] P. Faloutsos, M. van de Panne, and D. Terzopoulos, "Composable controllers for physics-based character animation," in *Proceedings of ACM SIGGRAPH 2001*, Los Angeles, CA, USA, Aug 2001, pp. 251–260.

[9] R. Tedrake, T. W. Zhang, , and H. S. Seung, "Stochastic policy gradient reinforcement learning on a simple 3d biped," in *IEEE Intelligent Robots and Systems*, 2004, pp. 2849–2854.

[10] S. Ramamoorthy and B. Kuipers, "Qualitative hybrid control of dynamic bipedal walking," in *Robotics:Science and Systems*, 2006.

[11] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng, "Learning cpg sensory feedback with policy gradient for biped locomotion for a full-body humanoid," in *National Conerence on Artificial Intelligence*, 2005, pp. 1267–1273.

[12] S. Vijayakumar, A. D'Souza, T. Shibata, J. Conradt, and S. Schaal, "Statistical learning for humanoid robots." *Auton. Robots*, vol. 12, no. 1, pp. 55–69, 2002.

[13] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 473–482, 2002.