Omnidirectional Humanoid Balance Control: Multiple Strategies for Reacting to a Push

KangKang Yin, Department of Computer Science
University of British Columbia, Canada
Email: kkyin@cs.ubc.ca
Michiel van de Panne, Department of Computer Science
University of British Columbia, Canada
Email: van@cs.ubc.ca

Abstract— **We develop and evaluate humanoid balance controllers that can recover from unexpected external perturbations of varying magnitudes in arbitrary directions. Balance strategies explored include ankle and hip strategies for in-place balance, as well as single-step, double-step, and multi-step balance recovery. Simulation results are provided for a 30 DOF humanoid.**

I. Introduction

The control of balance for humanoids is an important and largely unsolved problem. There are few control algorithms that support significant disturbances. We focus specifically on the problem of balance recovery for small and large pushes during quiescent stance. In this context, our contributions are: (1) we investigate and implement a list of control strategies for balance recovery from a wide range of unexpected perturbations; (2) we document the performance limits for the balance controllers that implement the above strategies; (3) we integrate these controllers as separate modes of a multi-strategy controller, together with limit cycle walking control.

II. Related Work

In robotics, various balance measures have been developed, including: ZMP (Zero Momentum Point [1]), FRI (Foot Rotation Indicator [2]), and ZRAM (Zero Rate of change of Angular Momentum [3]). With appropriate controller designs, these measures have been used to maintain balance for humanoid robots, most typically in the context of walking and relatively small disturbance [4], [5].

Linear and angular momenta are commonly used quantities in motion and balance control. From a reference momentum, voluntary whole body motion can be calculated [6]. Feedback methods using the AMPM model (Angular Momentum inducing inverted Pendulum Model) have also been proposed for bipeds to counteract external sagittal plane perturbations during walking [7]. With this model, the walking steps are unchanged from the original feedforward motion. In the results section we make specific performance comparisons with this technique and others, to the extent that this is possible.

In animation applications the goal is often to generate visually appealing animations instead of realizable simulations based on forward dynamics. As a result, kinematic approaches are commonly favored. Reactive motions to external pushes [8], [9], [10] can be animated using pre-captured reactive motions, momentum based inverse kinematics, and motion blending.

In computer animation using forward dynamics, the majority of work [11], [12], [13] uses PD controllers based on CoM regulation in order to balance in-place or land after airborne movements. Limit cycle control [14] uses local linear models to stabilize walking onto a limit cycle, which inspired the construction of our stepping controllers.

Recovery from large forward pushes during quiescent stance is an interesting case where a two-phase balance strategy becomes particularly effective [15]. In the reflex phase, the body deliberately moves away from the ideal posture to absorb a disturbance force and maintain controllability. The recovery phase attempts to restore the body to its original posture as the disturbance force subsides. The motions do not involve any stepping. In [16], an optimization procedure based on quadratic programming is combined with PD control, which can boost the magnitude of pushes a humanoid robot can cope with. The feet do not move, i.e., it is an in-place strategy.

The biomechanics and motor control community has studied human balance in considerable depth. Studies of how we control the equilibrium of the body in the face of gravity and environmental disturbances [17], [18], [19], [20] have led to the concept of *movement strategies*. Postural strategies describe general sensorimotor solutions to the control of posture, including not only muscle synergies but also movement patterns, joint torques, and contact forces. From human subject experiments, various balancing strategies have been observed in response to external perturbations, including an ankle strategy, a hip strategy, as well as change-of-support strategies such as stepping.

The *ankle strategy* uses distal to proximal muscle activation, primarily at the ankle and the knee. It is characterized by body sway resembling a single-segment-inverted pendulum and is typically elicited during small shifts on flat support surfaces or perturbations of CoM when the task requires maintenance of upright posture.

The *hip strategy* uses early proximal hip and trunk muscle activation. It is characterized by body sway resembling a double-segment inverted pendulum divided at the hip. It is typically elicited during perturbations that are large combined with a lack of a surface to support a step, on compliant support surfaces, or when the task requires a large or rapid shift in CoM.

The *stepping strategy* uses early activation of hip abductors and ankle co-contraction. It is characterized by asymmetrical loading and unloading of the legs to move the base of support under the falling CoM. This is typically elicited when there are no surface or instructional constraints, or when the perturbations are extremely large and in-place balance is not possible. Multiple steps may occur during balance recovery.

When examined in the above framework, [11], [12], [13] mainly use an ankle strategy and [15] employs a hip strategy. Our goal is to quantify how these strategies compare, as well as how they integrate with each other and with walking.

III. In-place Controllers

Balance control that does not involve any stepping is the most restrictive in terms of the magnitude of applied push.

A. Ankle Strategy

Ankle strategy is implemented as a proportional derivative (PD) controller that produces a virtual force $\mathbf{f}$ regulating CoM position $\mathbf{p}$ and velocity $\dot{\mathbf{p}}$.

$$\mathbf{f} = k_p(\mathbf{p}_{desired} - \mathbf{p}) - k_d\dot{\mathbf{p}} \tag{1}$$

The virtual torque needed from a joint (ankles, knees and hips) is defined by:

$$\tau_v = \mathbf{f} \times \mathbf{r} \tag{2}$$

where $\mathbf{r}$ is the vector from the CoM to the individual joint center. The virtual torque is then transformed by R, the matrix that relates the global coordinate system to the joint's coordinate system, into actual joint torques:

$$\tau_j = R\tau_v$$

The ankle strategy can be viewed as an equilibrium point tracking mechanism. It is limited by the fact that human feet are relatively small compared to the large and tall trunk. When the ZMP hits the foot support polygon boundary the ankle torques will lift the heel or the toe of the foot rather than providing the desired virtual force to push the CoM back into place.

B. Hip Strategy

From Equation 2, we see that the hip torques generated from the ankle strategy are very low, because $\|\mathbf{r}\|$ is small for the hips. Given that the hip joints have significantly greater torque generation capacity, we would like to use them more effectively in the overall balance strategy. From Section II and [15], the inspiration is that we can actively rotate the hip to induce angular momentum which causes the ZMP to remain within the foot support boundary. We implement this mechanism as simple linear synergy that co-activates the hips and the ankles:

$$\tau_{v-hip} = s\tau_{v-ankle}$$

In our experiments, we used $s = 3.0$.

To facilitate the self-induced rotation, the hip position gain (i.e., k_p, see Equation 3) is dropped to 20% of its original value during the yielding phase, and regains its strength as a linear function of time over $\beta_1 = 2.0$ seconds during the recovery phase. The virtual character transitions to the recovery phase when the CoM velocity rotates by more than 90 degrees, i.e., $\dot{\mathbf{p}}_t \cdot \dot{\mathbf{p}}_0 <= 0$, where $\dot{\mathbf{p}}_0$ is the original CoM velocity, and $\dot{\mathbf{p}}_t$ is the current CoM velocity. This separates the hip strategy into two distinct phases. The hip strategy is primarily effective for forward pushes. In our implementation, the hip strategy is only used for pushes that result in forward CoM motion in the sagittal plane.

IV. Stepping Controllers

We classify our stepping controllers into three types, based on how many steps the controller evokes before returning to a static upright posture.

- *Single step*: Only steps once, and recovers to an upright posture with a staggered foot stance.
- *double step*: Steps twice, and brings the trailing foot next to the stepping foot when returning to the upright posture.
- *multi-step*: Takes more than two steps to recover, with the number of steps being determined on the fly. The final quiescent stance pose is similar to the starting pose.

All of the stepping controllers share three phases:

- Phase 1: Change of support. In this phase, the controller decides where and how to lift-and-place the swing foot to change the foot support polygon. This is repeated multiple times for double-step and multi-step control.
- Phase 2: Reduction of momentum of the system. In this phase, the controller removes momentum either through stiffness and CoM velocity regulation in single and double stepping, or through up-vector regulation in multi stepping. The feet are already in their final state.
- Phase 3: Return to the upright posture. The controller steers the character back to an upright posture.

A. Where and How to Step

The basic control representation used for stepping and walking is the pose control graph (PCG), which is a type of finite state machine as shown in Figure 1. Each state P_i specifies a set of desired joint angles for the character, and the joints will be driven toward the desired angles through the use of PD controllers:

$$\tau = k_p(\theta_{desired} - \theta) - k_d\dot{\theta} \tag{3}$$

Figure 1 shows a typical PCG for stepping. It first lifts the left leg as shown on the left, and then steps down as shown on the right. State P_n can be the same as P_1 for a cyclic graph, usually for cyclic motions such as walking.

The state transition conditions in the PCG are time-based or sensor-based. In the PCG in Figure 1, we transition from pose P_1 to pose P_2 after 0.2s. Sensor based transition conditions we use in this work include foot contact and CoM velocity. For example, in the PCG for multistepping and walking, $P2$ will transition to P_3 (the symmetric counterpart of P_1) after the swing foot contacts the ground. CoM velocity is monitored to transition to Phase 3 of the stepping controllers. PCGs are designed manually, and are fixed for each type of stepping controllers. Designing a good PCG requires some trial-and-error, but is simplified by the fact that the PCG does not itself need to provide a balanced motion. Instead, it provides a base motion upon which control adjustments are then layered.

Control is provided by adding control variables to the swing hip, which thereby parameterizes the placement of the swing foot. The two Euler angles of the swing hip are θ_s for the sagittal angle and θ_l for the lateral angle. The control variables for the stepping thus consist of $\mathbf{c} = (\theta_s, \theta_l)^T$. We compose the final desired hip angles by adding the control variable hip angles to the hip angles from the base PCG.

Upon an unexpected push, people step using the leg which is unloaded by the push [21]. So if $\dot{\mathbf{p}} \cdot \mathbf{l} >= 0$, i.e., for a push to the left, we set the right leg to be the stepping leg, and left leg otherwise. $\dot{\mathbf{p}}$ is the CoM velocity at the beginning of a step, and $\mathbf{l}$ is the load line from the right foot to the left foot.

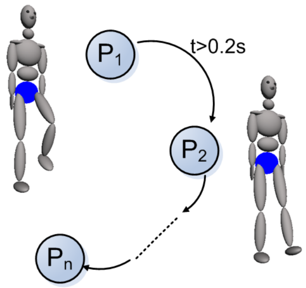


Fig. 1. A pose control graph represents the basic control for stepping and walking, and is a finite state machine. The states P_i specify desired poses. State transition are either time-based or sensor-based.

A key problem to solve when pushed is that of determining where to step to help recover balance. This should typically be a function of the current motion of the CoM. We apply an offline search procedure to compute the required solution and then use a function approximator to replace the search procedure in order to achieve efficient solutions that are usable in an online fashion. The search procedure is carried out in one of two ways:

- Automatic search and determination of controls on a dense grid, based on whether the balance is achieved or not. This is tractable because our control is only 2 dimensional.
- Intelligent search and determination of controls, based on a well-defined performance index. One example is the RV (regulation variable) from [14].

B. Single and Double Stepping

Our single and double stepping controllers share the same mechanism for Phase2 and Phase3, after touchdown of the swing foot:

- Reduction of momentum of the system. This is achieved in two ways. First, the position gains for the swing foot and knee are dropped to zero opon ground contact in order to reduce impact, and regains its original values linearly over time in $\beta_2 = 0.5$ seconds. Second, active torques are added to regulate the CoM velocity, i.e., Equation 1 with only the damping term.
- Return to the upright posture. When the velocity of the CoM is below a threshold $\gamma = 0.15m/s$, we deem it is now safe to start to return to the desired upright posture. A kinematic planner calculates desired ankle and hip angles, based on the current foot configuration and straightened knees. This new pose is added to the PCG to steer the character back to this posture over a duration of $\beta_3 = 0.5$ seconds.

These two phases after the change of support follow the philosophy of Section III-B and [15]. Reduction of momentum and recover of the upright posture can be conflicting goals in balance tasks upon large pushes. Separating them into two phases results in a more successful and robust balance controller.

C. Multi Stepping Controller

The single-step and double-step controllers both provide rapid stops, i.e., they bring the motion to rest as quickly as possible. Another strategy we investigate is a more gradual stop whereby the momentum caused by external forces is gradually dissipated. For this we employ the walking controller proposed in [14].

In [14], a PCG gives the basic open-loop controller for walking. Local linear models are then constructed through preview simulations to stabilize a set of regulation variables (RVs), and hence the walking itself, onto a limit cycle. Different open-loop controllers and different RVs can be used to achieve different styles as well as controlling the walking direction. For our purposes here, we only use a straight-line forward-walking controller, with the up-vector as the RVs. The up-vector measures the torso lean in the sagittal and frontal planes. To begin a walk, the torso typically leans forwards. Upon being pushed forward, the torso will lean forward substantially. For each step, instead of commanding the feedback controller to regulate the up-vector about a fixed target value as in a cyclic walking controller, we decrease it over time with the goal of achieving a gradual stop:

$$RV_n = \alpha * RV_{n-1} \qquad\qquad 0 < \alpha < 1$$

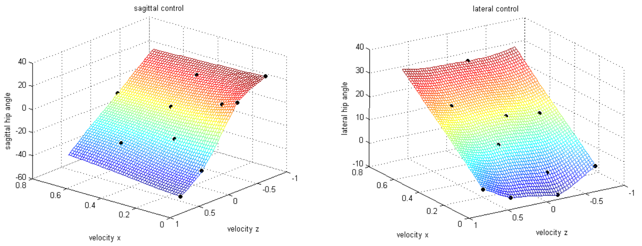
where n is the counter for the number of steps. For the results of Section VI, we set $\alpha = 0.9$. When the velocity of the CoM along the walking direction falls below a threshold $\gamma = 0.15m/s$, we execute Phase 3 as described in Section IV-B for the single and double stepping in order to achieve a full stop and recovery. This approach can also cope well with multiple pushes sustained during the gait.

V. Real-time Control

Phase 2 and Phase 3 of the stepping controllers are feedback mechanisms executable in an online fashion, as are the ankle and hip strategies for quiescent stance. Phase 1 of the stepping controllers, however, consists of feed-forward control of the swing hip angles. As detailed earlier, the controls have to be found through an off-line search process.

To make the stepping controllers more useful, we investigate how to make the first phase, i.e., finding where to step, an online procedure. We focus on the single stepping and double stepping strategies hereafter. Multi stepping, however, poses a more challenging task because the number of steps taken are undetermined beforehand.

Given a set of planar pushes of different directions and magnitudes $\mathbf{f}_i = (f_{ix}, f_{iz})^T$, we record the planar CoM velocities $\mathbf{v}_i = (v_{ix}, v_{iz})^T$ as the state variable, right after the push ends and just before any balance controller starts to execute. We use a Y-up coordinate system, and the y component of the velocity is discarded. Using one of the search techniques described in Section IV-A, we record the controls $\mathbf{c}_i = (\theta_{is}, \theta_{il})^T$. Using a small set of samples, we then construct a function $\mathbf{c}_i(\mathbf{v}_i)$ using a thin-spline function approximator. Figure 2 shows an example of the resulting control surface for a single-step controller. Given a new body

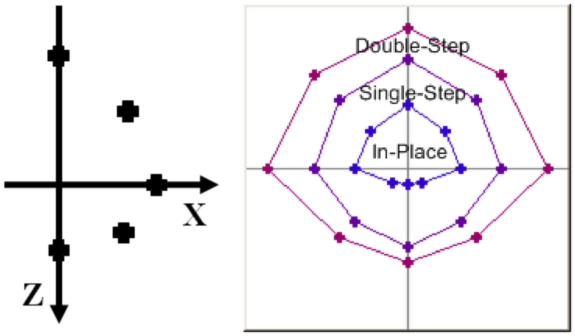


(a) (b)

Fig. 2. Thin plate interpolation of control points for single stepping. Left figure shows hip sagittal angle $\theta_s(\mathbf{v}_i)$; right figure shows hip lateral angle $\theta_l(\mathbf{v}_i)$.

state, the controls are readily predicted using the function approximator, which enables real-time online control.

The top middle column of Figure 4 illustrates the distribution of the control points $\mathbf{f}_i$ for single stepping. The axes of Figure 4 represent $\mathbf{f}_i$, which is the applied disturbance force as applied over a finite time period Δt. The magnitude, direction, and duration of the applied perturbation forces $\mathbf{f}_i$ are our independent variables in creating the push perturbations, and are naturally correlated to $\mathbf{v}_i$ according to $\mathbf{v}_i \approx \mathbf{f}_i \Delta t/m$.



(a) (b)

Fig. 3. (a) The initial control points are chosen from five representative directions. For each strategy, the maximum push in each direction that the controller can withstand is recorded. (b) The nest of the convex hulls of the control points of different strategies, excluding the multi-step strategy. The mixed real-time balance controller selects a specific controller based on which domain the user input push falls into.

To enable a compact control representation, we wish to use as few sample points $\mathbf{f}_i$ as possible. We first run simulations in batch mode while sampling along five representative directions as shown in Figure 3(a). We record the maximum push $\mathbf{f}$ a controller can deal with in each direction. We denote the five points for strategy S_i as $\mathbf{s}_{i_j}$, $j = 1, 2, 3, 4, 5$, and denote the convex hull of these points as $H(S_i)$. We only deal with half the plane (pushes with a component to the right) because the lateral symmetry of our character allows us to mirror the controls to cover the remaining directions. The convex hulls of each controller nest as shown in Figure 3(b). This suggests an integrated controller that can utilize multiple balance strategies by choosing an appropriate strategy based on the direction and magnitude of the observed push.

We order the strategies S_i by their balance recovery capabilities from low to high, i.e., in-place strategy, single stepping, double stepping. The domain of a particular strategy $\Omega(S_i)$ is defined as $\Omega(S_i) = H(S_i) - H(S_{i-1})$. The sample points $C(S_i)$ used to define a strategy initially consists of the points on its domain boundary, i.e., the 10 points $C(S_i) = \mathbf{s}_{i_j} \cup \mathbf{s}_{i-1_j}$. We desire that the controls interpolated from $C(S_i)$ work for all points $\mathbf{s} \in \Omega(S_i)$. To verify this, we run the controllers on a densely-sampled grid of test perturbations within the bounding box of its domain. We denote a successful balance recovery with a green '+', and a failure with a red '×', as shown in the lower row of Figure 4. The middle column is the result for the single-step controller. As we can see, the small number of boundary point samples are already enough for the controller to work for the whole domain.

For the double stepping controller there exist regions where the initial small set of boundary point samples is insufficient to guarantee success over the full domain that they enclose. This is perhaps because the sample points are widely scattered across a large region and that the double-stepping motion is simply more complex in nature. We remedy this by adding more sample points in the interior of $C(S_i)$. The centroids of failure region are chosen as points for placing additional samples. We do this iteratively until the controller works for the whole domain.

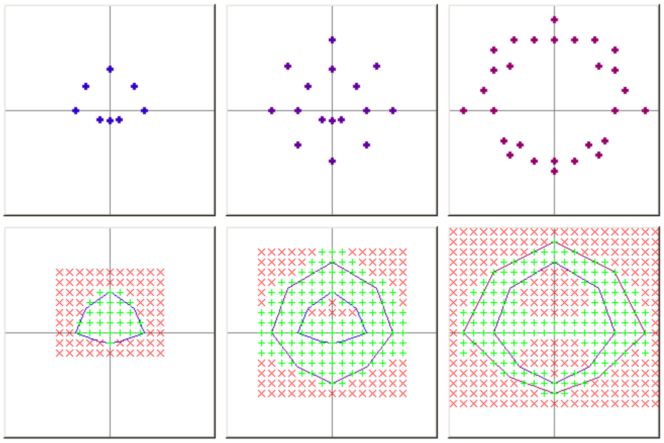


Fig. 4. Left column: in-place strategy. Middle column: single stepping. Right column: double stepping. Upper row: the control points for each strategy. Lower row: Nested polygon outlines the domain of a controller. Green '+' marks the successful region of the controller. Red '×' marks the failure region. The interval between the marks are 50 N. Thus perturbations range from -500 N to 500 N in each axial directions are tested for the double-step controller (i.e., the lower right figure), for example.

VI. Results

All controllers are tested with the simulation engine ODE (Open Dynamics Engine, http://ode.org/), on a full 3D model with 30 internal DoFs. Before the perturbations are applied, the virtual character adopts a standard quiescent stance pose that has the feet placed at shoulder width. The pushes we apply are impact forces of 0.2 seconds at the chest level. During the pushes, we delay the activation of our balance controller until the end of the $0.2s$ push in order to mimic the latency of human sensory-motor feedback loops [22]. If we allow instant feedback, we expect that the maximum pushes the controllers can endure will increase accordingly.

For all the controllers, the recoverable backward pushes are significantly less than those of the forward or sideways pushes. This conforms with our motion capture experiments with real humans, and we deem it to be a consequence of the anthropomorphic model. Also from the success of the relatively sparse control points for the stepping controllers, we conclude that our stepping controllers are smooth and robust, with respect to variations of push directions and magnitudes.

We now quantitatively compare our results to other results from the literature. [11] applied pushes below 150 N from different directions over a 0.25 second interval. In [16], 500 N forward push, 300 N backward push, and 200 N sideways push, each last 0.1 seconds are tested. The disturbance forces used in [15] is 300 N for 0.1 seconds. The perturbations reported in [7] range from 0.1 $kg \cdot m/s$ to 0.2 $kg \cdot m/s$ in linear momentum[1], and 12.0 $kg \cdot m^2/s$ to 24.0 $kg \cdot m^2/s$ in angular momentum.

A summary comparison of published results is shown in Table I. We note that a performance comparison of balance controllers based solely on the magnitude of pushes may be misleading, because the directions, locations, and durations of the pushes may be different, and the underlying kinematic and dynamic models are usually different. Since from a static state, the linear momentum injected into a system is $\Delta P = f\Delta t = m\Delta v$, we believe that the maximum CoM velocity caused by the push, i.e., the momentum normalized by the total mass, is possibly a better choice for performance comparison. Quantitatively, for forward pushes with our controller the maximum CoM velocity is $1.06 m/s$; for pushes sideways, the maximum velocity is $1.01 m/s$; for backward pushes, it is $0.75 m/s$.

paper	model	maximum pushes experimented		controller
		direction	ΔP ($kg \cdot m/s$)	
[11]	humanoid	forward	20	omnidirectional
		sideways	30	in-place only
		backward	16.25	
[16]	humanoid	forward	50	omnidirectional
		sideways	20	in-place only
		backward	30	
[15]	single leg	forward	30	single direction
	plus HAT			in-place only
[7]	humanoid	forward	0.2[1]	single direction
				during gait
ours	humanoid	forward	90	omnidirectional
		sideways	90	multiple strategies
		backward	60	

TABLE I

SUMMARY OF WORK DEALING WITH PUSH RECOVERY.

We also tested the robustness of our controllers with respect to model variations, such as limb mass and limb length. We can decrease the mass of every limb up to 5% and the control for the single stepping works without any changes. If we decrease the mass by 10%, one out of the ten control points fails, and small adjustments (¡ 2 degrees) are needed to make them work again. We also tried to increase the shank length, for which the unaltered control can support a 6% shank length increase.

Figure 5 shows some motions from simulations resulting from pushes of various magnitudes and directions. Corresponding videos are available from http://www.cs.ubc.ca/∼kkyin/animation.

VII. CONCLUSION AND FUTURE WORK

We have presented a first demonstration and performance characterization of a humanoid balance controller that is capable of selecting among multiple strategies in order to react to small and large pushes from any direction. The stepping strategies rely on learning a model of where to step as a function of the CoM velocity resulting from the push. The resulting balance controllers work in real-time to maintain the balance of a 30-DOF simulation of a humanoid. The domains of the various control strategies are characterized in detail.

In the future, we wish to deal with a number of outstanding issues. The PCGs that provide the base stepping motion could be constructed automatically from motion capture data. Lateral pushes can currently cause the swing leg to collide with the stance leg. This behavior can be predicted and corrected for. Additional strategies can be added, such as an arm rotation strategy. The demonstrated balancing skills should be integrated with models of other motor skills.

ACKNOWLEDGMENT

The authors would like to thank Kevin Loken for various discussions during the course of this project, and proofreading of this paper.

REFERENCES

[1] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *IEEE Transactions on Biomedical Engineering*, vol. 16, pp. 1–6, 1969.

[2] A. Goswami, "Postural stability of biped robots and the foot rotation indicator (fri) point," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, 1999.

[3] A. Goswami and V. Kallem, "Rate of change of angular momentum and balance maintenance of biped robots," in *IEEE Intern. Conf. on Robotics and Automation*. IEEE, 2004, pp. 3785–3790.

[4] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue, "Autobalancer: An online dynamic balance compensation scheme for humanoid robots," in *Fourth Int. Workshop on Algorithmic Foundations on Robotics*, 2000.

[5] Y. Okumura, T. Tawara, K. Endo, T. Furuta, and M. Shimizu, "Realtime zmp compensation for biped walking robot using adaptive inertia force control," in *International Conference on intelligent Robots and Systems*, 2003.

[6] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *IEEE Intern. Conf. on Intelligent Robots and Systems*. IEEE, 2003, pp. 1644–1650.

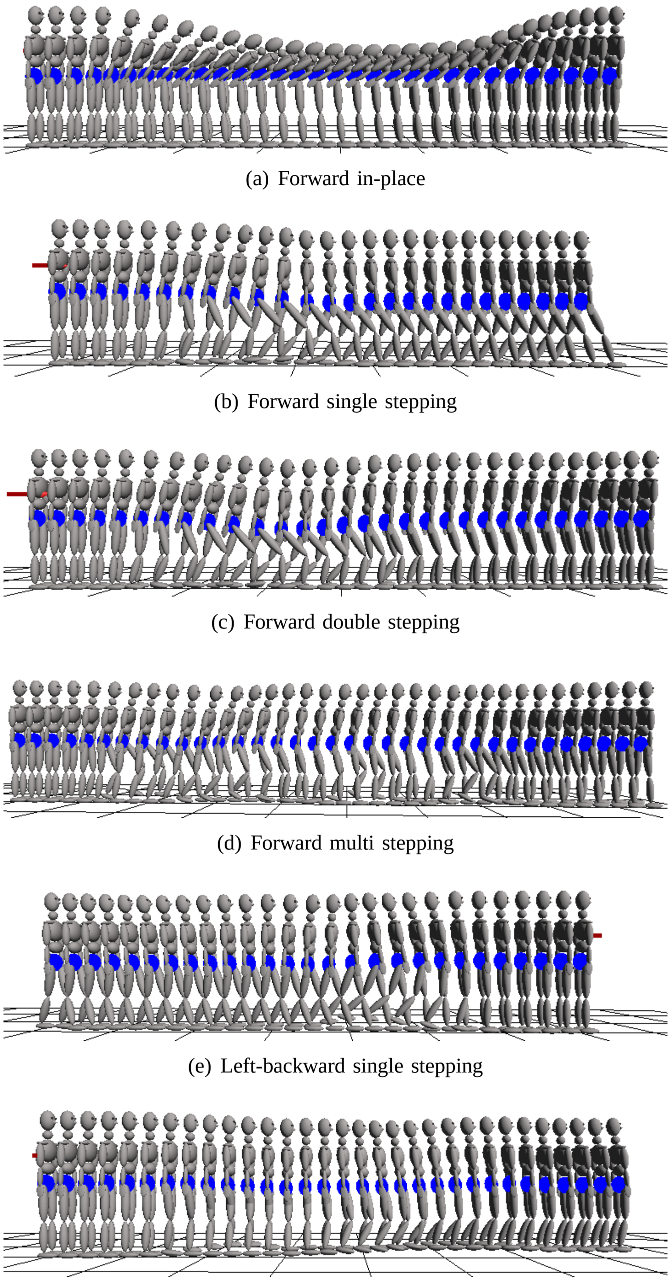
[7] T. Komura, H. Leung, S. Kudoh, and J. Kuffner, "A feedback controller for biped humanoids that can counteract large perturbations during gait," in *International Conference on Robotics and Automation*, 2005, pp. 2001–2007.

[8] O. Arikan, D. A. Forsyth, and J. F. O'Brien, "Pushing people around," in *Symposium on Computer Animation*, 2005.

[9] K. Yin, D. K. Pai, and M. van de Panne, "Data-driven interactive balancing behaviors," in *The Thirteenth Pacific Conference on Computer Graphics and Applications*, 2005.

[10] T. Komura, E. S. L. Ho, and R. W. H.Lau, "Animating reactive motion using momentum-based inverse kinematics," *Computer Animation and Virtual Worlds*, vol. 16, no. 3-4, pp. 213–223, 2005.

[1] Judging from the magnitudes, there may be an error in the reported numbers.



(a) Forward in-place

(b) Forward single stepping

(c) Forward double stepping

(d) Forward multi stepping

(e) Left-backward single stepping

(f) Leftward double stepping

Fig. 5. Balance animation upon pushes of different magnitudes and directions.

[11] W. L. Wooten, “Simulation of leaping, tumbling, landing, and balancing humans,” Ph.D. dissertation, Georgia Institute of Technology, 1998.

[12] V. B. Zordan and J. K. Hodgins, “Motion capture-driven simulations that hit and react,” in *ACM SIGGRAPH Symposium on Computer Animation*, July 2002, pp. 89–96.

[13] 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.

[14] J. Laszlo, M. van de Panne, and E. Fiume, “Limit cycle control and its application to the animation of balancing and walking,” in *SIGGRAPH ’96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM Press, 1996, pp. 155–162.

[15] M. Abdallah and A. Goswami, “A biomechanically motivated two-phase strategy for biped upright balance control,” in *International Conference on Robotics and Automation*, 2005, pp. 2008–2013.

[16] S. Kudoh, T. Komura, and K. Ikeuchi, “The dynamic postural adjustment with the quadratic programming method,” in *International Conference on Intelligent Robots and Systems*, 2002.

[17] F. B. Horak, S. M. Henry, and A. Shumway-Cook, “Postural perturbations: new insights for treatment of balance disorders,” *Physical Therapy*, vol. 77, no. 5, pp. 517–534, 1997.

[18] A. Shumway-Cook and M. H. Woollacott, *Motor Control: theory and practical applications*, 2nd ed. Lippincott Williams & Wilkins, 2001.

[19] B. E. Maki, W. E. McIlroy, and G. R. Fernie, “Change-in-support reactions for balance recovery,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 2, pp. 20–26, 2003.

[20] A. E. Patla, “Strategies for dynamic stability during adaptive human locomotion,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 2, pp. 48–52, 2003.

[21] B. E. Maki, W. E. McIlroy, and S. D. Perry, “Influence of lateral destabilization on compensatory stepping responses,” *J Biomech*, vol. 29, no. 3, pp. 343–353, 1996.

[22] M. Kawato, “Internal models for motor control and trajectory planning,” *Current Opinion in Neurobiology*, vol. 9, pp. 718–727, 1999.