Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2011
August 29-31, 2011, Washington, DC, USA

DETC2011/MESA-47153

FALL ON BACKPACK: DAMAGE MINIMIZING HUMANOID FALL ON TARGETED BODY SEGMENT USING MOMENTUM CONTROL

Sung-Hee Lee
School of Information and Communications
Gwangju Institute of Science and Technology
Gwangju, South Korea 500-712
Email: shl@gist.ac.kr

Ambarish Goswami*
Honda Research Institute USA
Mountain View, CA 94043
Email: agoswami@honda-ri.com

ABSTRACT

Safety and robustness will become critical issues when humanoid robots start sharing human environments in the future. In physically interactive human environments, a catastrophic fall is the main threat to safety and smooth operation of humanoid robots, and thus it is critical to explore how to manage an unavoidable fall of humanoids.

This paper deals with the problem of reducing the impact damage to a robot associated with a fall. A common approach is to employ damage-resistant design and apply impact-absorbing material to robot limbs, such as the backpack and knee, that are particularly prone to fall related impacts. In this paper, we select the backpack to be the most preferred body segment to experience an impact. We proceed to propose a control strategy that attempts to re-orient the robot *during the fall* such that it impacts the ground with its backpack. We show that the robot can fall on the backpack even when it starts falling sideways. This is achieved by utilizing dynamic coupling, i.e., by rotating the swing leg aiming to generate spin rotation of the trunk (backpack), and by rotating the trunk backward to drive the trunk to touch down with the backpack. The planning and control algorithms for fall are demonstrated in simulation.

INTRODUCTION

As humanoid robots start to migrate from controlled laboratory environments to free and physically interactive surroundings containing objects and people, the issues of safety and robust operation of these robots will demand major attention from the research community. A catastrophic fall is perhaps the gravest threat to the safety and security of these robots and their surroundings. Yet, despite the stringent control that is imposed on every humanoid movement, fall remains an uncontrolled, understudied, and basically overlooked aspect of humanoid technology.

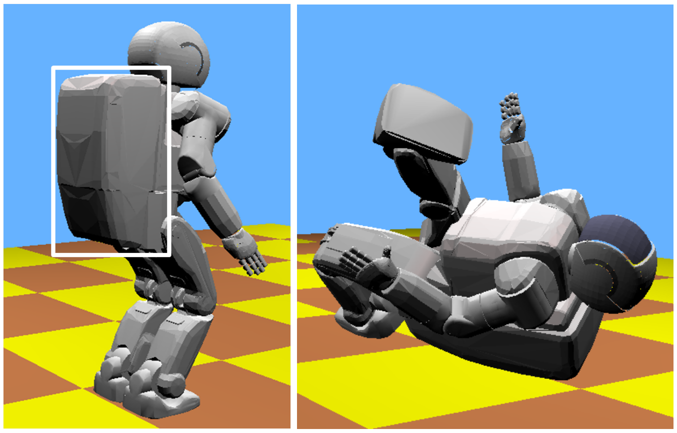
Because of its complex dynamics and lack of well-defined theoretical tools, one is tempted to completely ignore the treatment of a humanoid fall. This, however, does nothing to reduce the chances of fall, but rather makes the effects of fall unpredictable and potentially more harmful. In a comparable situation involving automobiles, we have learned that crash studies can significantly improve the "crashworthiness" of a car by increasing the safety of both car occupants and outside pedestrians. Inspired by this, we deliberately focus our attention to the phenomenon of humanoid fall and attempt to develop a comprehensive control strategy to deal with this undesired and catastrophic event.

During an accidental fall the humanoid controller may have two primary, and distinctly different, objectives: a) self-damage minimization and b) minimization of damage to other objects or people. If during the fall the robot can hit nearby objects or persons, its primary objective would be to prevent this from happening. The robot may try to achieve this by means of changing its default fall direction, as we have reported [1, 2]. If, however, the fall occurs in an open space, a self-damage minimization strategy

*Address all correspondence to this author.

Copyright © 2011 by ASME

can attempt to reduce the harmful effects of the ground impact.



(a) (b)

FIGURE 1. Backpack is rigidly attached to the back of the trunk (left) and we aim to control a falling robot to touch down with a specific body part, in our case the backpack (right).

One practical approach to reducing impact related damages is to employ better design and apply impact-absorbing materials to robot parts that are frequently prone to fall-related impacts. One can identify a number of body segments, such as knee, hip, hand, head, etc., that are more likely to contact the floor in case of a fall. For many humanoid robots, the backpack is a segment that, due to its design and 3D profile, would come into frequent contact with the ground if the robot were to fall. The backpack, which is an integral part of the humanoid trunk, can be properly designed to prevent the effect of the impact propagating to other, more fragile, components. By virtue of its special design, the backpack, akin to an automobile bumper, becomes better-equipped to face an impact compared to other parts. The challenge then is to ensure that the backpack is the part of choice, more than other parts of the robot, with which the robot ought to touch the ground if it were to fall.

Precisely with this objective this paper proposes a control strategy that attempts to re-orient the falling robot such that it impacts the ground at the backpack. This is a difficult task because the underactuation of the falling robot prevents the controller from achieving a certain orientation of a body part using only kinematic relation with other body parts. Rather, the movement of a body part can only be understood through the consideration of dynamic coupling with other body parts.

When the ground reaction force (GRF) and the gravity are the only external forces applied to the robot, the angular momentum about the *lean line*, the line connecting the center of pressure (CoP) and the center of mass (CoM), of a falling robot is conserved because all external forces have zero moment-arms from the lean line.[1] This behavior is similar to that of a rotating top where the angular momentum about the spin axis is conserved. For our model, the precession effect that can be observed in a rotating top is negligible due to a low angular velocity around the spin axis for normal falls. Therefore the change of angular velocity of the trunk about the lean line can only be achieved by rotating other body parts in the opposite direction. On the other hand, due to the ground contact, the angular momentum in the other directions is not preserved and can be changed. In this paper, we introduce an inverse dynamics-based control algorithm that controls the joint motion and the GRF to achieve the desired touchdown orientation of the trunk at touchdown time using momentum-based dynamic coupling of the links.

Time is a premium during the occurrence of a fall; single rigid body models of human-sized humanoid robots indicate that a fall from the vertical upright stationary configuration due to a mild push takes about 800-900ms. In typical situations the time to fall may be even shorter, and there is no opportunity for elaborate planning or time-consuming control. Yet, simulation and experimental results indicate that meaningful modification to the default fall behavior can be imparted to minimize damage to the robot or to the environment.

Reported work in the area of humanoid fall is relatively rare. A few recent papers reported on the damage minimization aspect of humanoid fall. In their body of work Fujiwara et al. [3–8] proposed martial arts type motion for damage reduction, computed optimal falling motions using minimum impact and angular momentum, and fabricated special hardware for fall damage study. Ishida et al. employed servo loop gain shift to reduce shock due to fall [9]. Ogata et al. proposed online planning techniques for the trajectory of the center of mass of a falling robot to reduce impact [10, 11] and del Solar et al. manually tuned falling motion of soccer robots [12]. Fall damage minimization is obviously of natural interest in biomechanics [13–15].

Our work is different from the previous work in that our technique attempts to fall on a specific body part that is more resistive to impact than other body parts whereas most of the previous work that dealt with impact reduction did not prioritize body parts.

Sensor noise and modeling errors can negatively affect the accuracy and robustness of the presented controller. Given the highly non-linear nature of the problem, such effects are difficult to estimate and predict. Eventually hardware tests can justify the values of such control strategies. However, the effect of a physical fall on the system hardware integrity, even in a controlled environment, can be unpredictable, or worse, severely damaging. Therefore one may not wish to undertake an exhaustive test in hardware. Consequently, our dependence on simulation should remain strong.

[1] We assume that the support foot and the ground make a point contact and the rotational friction between the foot and the ground is negligible.

Copyright © 2011 by ASME

FALL CONTROL FRAMEWORK

The problem addressed in this paper is quite unique in many respects. It is instructive to analyze the special characteristics of this problem and understand its constraints and limitations.

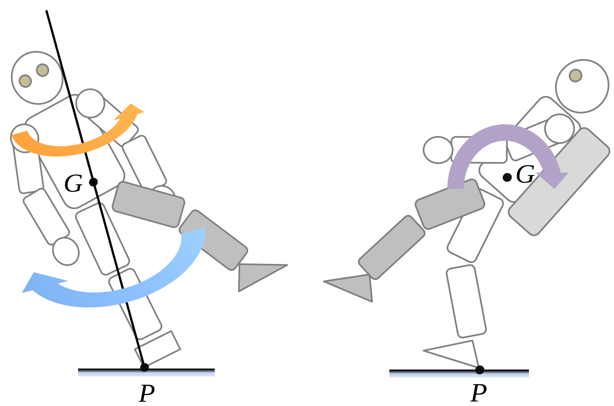
The control objective is that a falling humanoid should preferentially make the first contact with the ground using a targeted body part, in this case the backpack. This objective can be divided into two parts, the first is to re-orient the trunk so that the backpack faces the ground at touchdown, although we do not know the exact time of touchdown. The second is to ensure that all other body parts, except the feet, are above the ground during touchdown. The second objective implies that the backpack is the first body part to touch the ground and is mainly meant to keep the arms out of the way. To deal with this we take the simple step of maintaining or returning the arms to their default pose of keeping the hands naturally positioned in front of the pelvis. This way the arms will be located above the ground when the robot touches the ground with the backpack.

In order to re-orient the trunk so that the robot touches the ground with the backpack, we may specify the desired world-frame orientation of the backpack at the instant of touchdown and try to achieve it purely based on the kinematic relation of the links. However, the problem is much more involved.

The previous description conveniently hides the fact that the falling (tipping) robot is underactuated and a kinematic control cannot account for the involved dynamics. It is true that we have a desired orientation with which we wish the backpack to make contact with the ground during touchdown. However, we have no idea of the time the robot takes to touchdown. Moreover, even if we knew the time-to-fall, an attempt to kinematically reorient the backpack, might not achieve the desired result. Due to underactuation a certain rotation of the backpack may result in a different rotation in another segment, and completely modify the trunk orientation in complex non-intuitive ways.

Therefore, it is essential to take into account the dynamic coupling among the body parts of the falling robot: by moving other parts of the robot, we will attempt to achieve the desired orientation of the trunk. The arms of our robot model are relatively lightweight, and they do not affect the whole body dynamics very much. Therefore, we will use only the swing leg, which is relatively heavier, for this purpose. By rotating the swing leg, we will attempt to indirectly control the trunk orientation using inertial effects. We do not use the support leg for this purpose. This choice is due to the fact that the support leg has short moment arm from the lean line (see Fig. 2(a)), so it has only a small effect on the rotational dynamics about the lean line, and that our robot model has a fairly limited range of hip joint motion about the vertical axis, which restricts its ability to generate a large momentum.

If the robot is able to freely rotate about the spin axis, it will be able to fall on the backpack regardless of the initial fall direction. The lean line, described before, plays an important role as the spin axis. The gravity and the GRF do not create torques about the spin axis as they both intersect it. Consequently, the corresponding component of the angular momentum is preserved. This means that the only way to control the spin angle of the trunk is by rotating some other member, such as the swing leg, about the spin axis. However, because the swing leg rotation is constrained by joint limits, we may not attain the desired trunk orientation only through this means. To increase the chance of falling on the backpack, we take the additional measure of bending the trunk backward as will be detailed later. Fig. 2 illustrates these strategies.



(a) Spinning with swing leg (b) Bending the trunk backward

FIGURE 2. The strategies for falling on backpack are illustrated. (a) By counter-rotating the swing leg, the robot controls the spin. (b) Additionally, the robot bends the trunk backward to increase the possibility to touch down with the backpack. G and P denote CoM and CoP, respectively.

Fall control consists of planning and execution steps. When a robot detects an impending inevitable fall, it first plans how to fall. In order to touch down with the backpack, the controller determines the desired rotational velocity of the trunk given the current state of the robot. The planning is performed multiple times during the fall to update the desired velocity while it is falling. At each control time step, the robot determines the joint acceleration and torques to realize the desired falling motion.

We assume that we can either compute or measure the position and orientation of the trunk and its linear and angular velocities as well as the joint angles and velocities using sensors such as gyroscopes, accelerometers, and joint encoders. We also assume that the feet are equipped with force/torque sensors to measure CoP.

Copyright © 2011 by ASME

PLANNING FALL CONTROL

Since spin control is essential to our fall controller, we will define the *spin frame* and describe the centroidal dynamics of the humanoid with respect to this reference frame. As shown in Fig. 3, the spin frame is located at the CoM. Its axes, $\{t,b,s\}$, correspond to frontal, bending, and spin, respectively. The spin axis s coincides with the lean line GP. The frontal axis t is given by the intersection of the sagittal plane of the trunk and the plane normal to s (translucent circle). The torso bending direction b is given by $s \times t$.

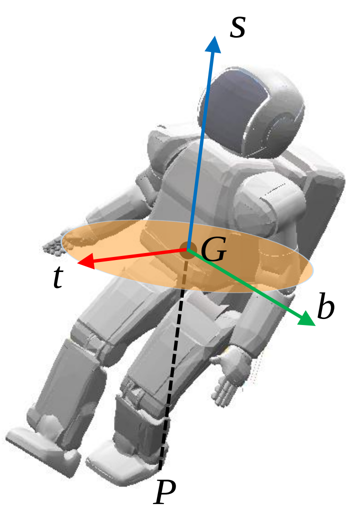


FIGURE 3. The *Spin frame* is located at the CoM and is defined by the axes $\{t,b,s\}$. The spin axis s coincides with the lean line $\overrightarrow{PG}$. Frontal axis t is the intersection of the sagittal plane of the trunk and the plane normal to s (translucent circle). The bending direction b is given by $s \times t$.

The centroidal angular momentum k, which is the aggregate angular momentum of a humanoid robot projected at its CoM, and the generalized velocity vector $\dot{q}$ have the linear relations as follows [16]:

$$k = K\dot{q}, \tag{1}$$

where k is a 3×1 vector and, for a robot with n joints, $\dot{q} \in \mathbb{R}^{n+6}$ is the generalized velocity vector and $K \in \mathbb{R}^{3\times(n+6)}$ transforms the generalized velocity vector to angular momentum. K is called the centroidal momentum matrix if k refers to the angular momentum about CoM [16].

To extract the motion of the link of interest, the trunk in our case, we divide $\dot{q} = (\omega, \upsilon, \dot{\theta})$ into the angular and linear velocities of the trunk (ω, υ) and the joint velocity vector $\dot{\theta} \in \mathbb{R}^n$. Then Eqn. (1) is expanded as:

$$k = K_w\omega + K_v\upsilon + K_\theta\dot{\theta}, \tag{2}$$

where K_w, K_v, and K_θ are sub-matrices of K corresponding to ω, υ, and $\dot{\theta}$, respectively.

If we express k, ω, and υ with respect to the spin frame (or, more specifically, a fixed frame that coincides with the spin frame instantaneously), the linear term vanishes because the spin frame is located at the CoM and the linear velocity of the trunk υ does not affect k:

$$k = K_w\omega + K_\theta\dot{\theta}, \tag{3}$$

where K_w is the locked rotational inertia of the robot, given by a symmetric positive definite matrix. Later, in the planning step we will pay attention to the spin component k_s of $k = (k_t, k_b, k_s)$.

Touchdown Time and Trunk Orientation Estimation

In order to control the orientation of the backpack so that it hits the ground first, it is important to know when the robot will collide with the ground. Although the exact time of touchdown depends on the movement of the robot, we can get a rough estimate by simulating a reduced model of the robot. For faster computation, we assume that all the joints of the robot are locked, such that the entire robot is treated as a single rigid body, and simulate the motion of the rigid body until it collides with the ground. Specifically, the rigid body is set to rotate about the robot CoP with the initial rotational velocity being the same as that of the robot trunk. We perform the dynamic simulation until the height of CoM falls below a threshold h_t, approximating the collision between the robot and the ground, and estimate the duration of fall t_f. We set h_t to about one third of the height of CoM of the robot standing with the default pose.

Also, we measure the orientation of the backpack at touchdown. This would be the backpack orientation if the robot did not take any action i.e., it locked all its joints during fall. This is the starting point of our fall control.

Desired Trunk Angular Velocity Estimation

We do not care about the specific collision point on the backpack, so the rotation of the trunk in the sagittal plane is not taken into account (Fig. 4). Therefore, we deal with only the vector n normal to the backpack (n points backward from the trunk) and aim to make n such that the plane made by GP and n is normal to the ground (n' in Fig. 5), which indicates that the backpack is facing the ground.

Given the estimated normal vector n at the time of touchdown, we determine the spin angle ϕ that is required to rotate n

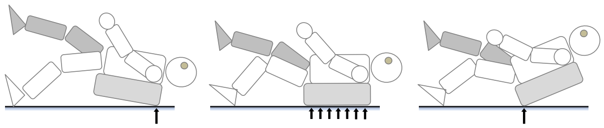


FIGURE 4. Various configurations of the backpack when a robot successfully touches down with the backpack. Flat contact (middle) may reduce impact at collision, but it may not be achieved in practice because normally backpack has a curved contour. We do not care which point of backpack collides with the ground.

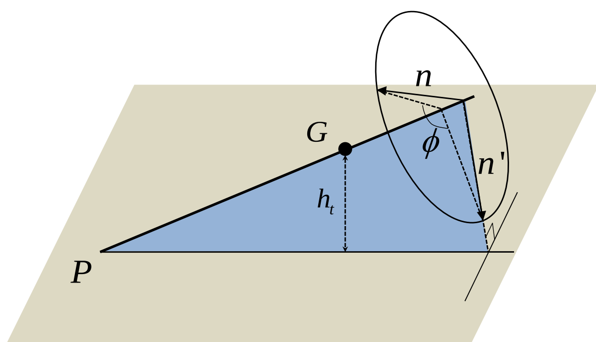


FIGURE 5. A robot model reduced to a single rigid body (i.e., all the joints are locked) is simulated until its CoM reaches a certain height h_t then the time to fall t_f and the vector n normal to the backpack is measured.

about the spin direction to the desired vector n' (Fig. 5). Then the desired spin velocity $\omega_{s,d}$ is determined as

$$\omega_{s,d} = \omega_{s,0} + \frac{\phi}{t_f}, \tag{4}$$

where $\omega_{s,0}$ is the initial value of the spin component of angular velocity of the trunk at the preview stage, which remains constant during the simulation, and t_f is the estimated duration of fall that was computed previously. $\omega_{s,d}$ specifies the spin velocity that will enable the robot to fall on the backpack, but it may not be physically realizable due to limits on joint angles and velocities of the swing leg. Therefore, we need to determine its *admissible* value. To this end, we use the third row of the momentum relation of Eqn. (3).

$$k_s = K_{w(3,1)}\omega_t + K_{w(3,2)}\omega_b + K_{w(3,3)}\omega_s + K_\theta\dot{\theta} \tag{5}$$

Recall that the spin component of the angular momentum k_s is conserved because no external forces induce torques about the spin axis. To simplify Eqn. (5), we will make the following approximations. 1) The contributions of frontal and bending components of angular velocity to k_s are negligible, i.e., $K_{w(3,1)}w_t \approx 0$, $K_{w(3,2)}w_b \approx 0$, and 2) only the joint velocities of the hip and knee joints of the swing leg are important with respect to the spinning. Then Eqn. (5) is approximated as

$$k_s \approx K_{w(3,3)}\omega_s + K_{swg}\dot{\theta}_{swg}, \tag{6}$$

where $\dot{\theta}_{swg}$ is the joint velocity vector of the swing leg. Then the possible range of ω_s is determined from the constraints on joint angles and velocities of the swing leg.

$$\omega_s^l \le \omega_s \le \omega_s^u, \tag{7}$$

where

$$\omega_s^l = \min \frac{k_s - K_{swg}\dot{\theta}_{swg}}{K_{w(3,3)}} \tag{8}$$

$$\omega_s^u = \max \frac{k_s - K_{swg}\dot{\theta}_{swg}}{K_{w(3,3)}} \tag{9}$$

The extrema of $\dot{\theta}_{swg}$ of each joint are determined from the lower and upper limits on joint angle and velocity as well as the current joint angle. For example, the maximum velocity that a joint $\dot{\theta}_i$ can have is set to be $\min(\dot{\theta}_i^u, (\theta_i^u - \theta_i)/t_f)$ where θ_i^u and $\dot{\theta}_i^u$ are the upper limits for joint angle and velocity. If the desired spin velocity is within the range, the admissible spin velocity $\omega_{s,a}$ can be set to the desired value. Otherwise, it is clipped to either ω_s^l or ω_s^u.

Next, we determine the desired bending velocity. If the desired spin velocity is admissible, the robot will be able to touchdown with the backpack without rotating the trunk backward. If the difference between the desired value and the admissible value of the spin velocity is large, the robot would have to rotate backward to increase the chance of falling on the backpack. Using this rationale, we use a heuristic method to determine the backward-bending velocity to be proportional to the difference of the $\omega_{s,a}$ and $\omega_{s,d}$:

$$\omega_b = -c|\omega_{s,a} - \omega_{s,d}|, \tag{10}$$

where c is a scaling factor. Following some trial cases, we set $c = 3$. Altogether, the desired angular velocity of the trunk is set as follows:

$$\omega_d = (0, \omega_b, \omega_{s,a}) \tag{11}$$

Copyright © 2011 by ASME

The first component of ω_d is set to zero to reduce the rotational component which is not particularly related with our control strategy.

EXECUTING FALL CONTROL

Given the desired angular velocity in Eqn. (11) of the trunk computed in the planning step, the fall controller controls the joint torques to realize the desired angular velocity.

To this end, we first derive the governing equations that relate the angular acceleration of the trunk, joint accelerations, and GRF. From this relation, we determine the desired joint accelerations and GRF that will drive the robot to have the desired angular velocity of the trunk, followed by computing necessary joint torques to realize the desired joint accelerations.

Governing Equations

The relation between GRF at the contact point and the rate of change of net spatial momenta are as follows:

$$\dot{k} = K\ddot{q} + \dot{K}\dot{q} = GP \times f \tag{12}$$

$$\dot{l} = L\ddot{q} + \dot{L}\dot{q} = f + mg, \tag{13}$$

where f is GRF, l is linear momentum, m is the total mass, and g is the gravity vector. Recall that GP is the lean line which connects the CoM and the CoP. Eqn. (12) is obtained by differentiating Eqn. (1). The matrix $L \in \mathbb{R}^{3\times(n+6)}$ transforms the generalized velocity vector to linear momentum, i.e., $l = L\dot{q}$.

Additionally, if the robot is to keep the contact with the ground, the following kinematic constraint relation should be maintained:

$$\dot{P} = J\dot{q} = 0, \tag{14}$$

where P is the CoP location vector and $J = dP/dq$. We will use its time differentiation:

$$0 = J\ddot{q} + \dot{J}\dot{q} \tag{15}$$

As in Eqn. (2), let us divide $\ddot{q}$ into the angular and linear accelerations of the base frame $(\dot{\omega}, \dot{\upsilon})$ and the joint accelerations $\ddot{\theta}$. Then Eqns. (12), (13), and (15) are rewritten respectively as:

$$K_w\dot{\omega} + K_v\dot{\upsilon} + K_\theta\ddot{\theta} + \dot{K}\dot{q} = GP \times f \tag{16}$$

$$L_w\dot{\omega} + L_v\dot{\upsilon} + L_\theta\ddot{\theta} + \dot{L}\dot{q} = f + mg \tag{17}$$

$$J_w\dot{\omega} + J_v\dot{\upsilon} + J_\theta\ddot{\theta} + \dot{J}\dot{q} = 0 \tag{18}$$

If we express ω, υ, and Eqns. (16)-(18) with respect to the spin frame, the angular and linear accelerations are decoupled, i.e., $K_v = L_w = 0$. Also, J_v and L_v become identity matrices, and $J_w = -[GP]$.[2] This gives us:

$$K_w\dot{\omega} + K_\theta\ddot{\theta} + \dot{K}\dot{q} = GP \times f \tag{19}$$

$$m\dot{\upsilon} + L_\theta\ddot{\theta} + \dot{L}\dot{q} = f + mg \tag{20}$$

$$-GP \times \dot{\omega} + \dot{\upsilon} + J_\theta\ddot{\theta} + \dot{J}\dot{q} = 0 \tag{21}$$

Since we are not interested in the linear motion of the trunk, we will eliminate $\dot{\upsilon}$ by substituting Eqn. (21) into Eqn. (20) to get further simplified equations.

$$GP \times \dot{\omega} + \left(\frac{1}{m}L_\theta - J_\theta\right)\ddot{\theta} + \frac{1}{m}\dot{L}\dot{q} - \dot{J}\dot{q} = \frac{1}{m}f + g \tag{22}$$

Eqns. (19) and (22) show the dynamic coupling among the angular acceleration of the trunk, joint accelerations, and GRF under kinematic constraints. Obviously, the joint accelerations $\ddot{\theta}$ completely determine both $\dot{\omega}$ and f and it is possible to eliminate f to express $\dot{\omega}$ in terms of only $\ddot{\theta}$. However, we will include f in the governing equations because f conveniently describes the constraints (contact force should be inside friction cone) as well as the global linear motion of the robot, i.e., $\ddot{r}_G = f/m + g$ where r_G is CoM.

Fall Control Command

Our goal is to determine $\ddot{\theta}$ and f such that desired orientation of the trunk is achieved at the time of ground impact. Let us assume that we have the desired value of the rotational acceleration of the trunk $\dot{\omega}_d$ as well as the desired joint accelerations $\ddot{\theta}_d$ and GRF f_d. Given the desired values, determining $\ddot{q}$ and f from Eqns. (19) and (22) can be described as a constrained least-squares problem:

$$\begin{gathered}\min(1-w)||Ax-b||^2 + w||x-x_d||^2 \\ \text{s.t. joint limit constraints,}\end{gathered} \tag{23}$$

where

$$x^T = (\dot{\omega}^T, \ddot{\theta}^T, f^T) \tag{24}$$

$$A = \begin{bmatrix} K_w & K_\theta & -[GP] \\ [GP] & \left(\frac{L_\theta}{m} - J_\theta\right) & -\frac{1}{m}I \end{bmatrix} \tag{25}$$

$$b = \begin{bmatrix} -\dot{K}\dot{q} \\ -\frac{1}{m}\dot{L}\dot{q} + \dot{J}\dot{q} + g \end{bmatrix} \tag{26}$$

[2] $[\cdot]$ denotes a 3×3 skew symmetric matrix, i.e., $[\omega]\upsilon = \omega \times \upsilon$.

Copyright © 2011 by ASME

$x_d^T = (\dot{\omega}_d^T, \ddot{\theta}_d^T, f_d^T)$ is the desired value for x. Since $Ax = b$ is an under-determined problem, we include an additional cost function $||x - x_d||^2$ to find the solution close to the desired values x_d, with a small weighting factor w (= 0.01 in our experiment). We enforce inequality constraints on $\ddot{\theta}$ to satisfy joint limits. It is done rather heuristically, e.g., by giving smaller upper bound on a joint acceleration if the joint gets closer the upper limit.

The details of setting x_d is as follows. At each control time step, the desired angular acceleration of the trunk $\dot{\omega}_d$ is set as:

$$\dot{\omega}_d = \gamma(\omega_d - \omega), \tag{27}$$

where γ is a feedback gain parameter. We set $\gamma = 20$ in our experiment through trial and error.

Next, we need to determine $\ddot{\theta}_d$ and f_d. We set $\ddot{\theta}_d$ for the arm joints such that the arms follow the prescribed motion, in our case maintaining the default arm pose, and set $\ddot{\theta}_d = 0$ for the leg joints to give a preference to minimal leg movement.

Setting f_d is a bit more involved and has significant effect on the behavior of the fall controller. Setting f_d too big will make the robot jump into the air, whereas setting it too small will make the robot collapse even before achieving our control goal. Our approach is to set f_d such that CoM of the robot moves as if the robot were a falling 3D point-mass pendulum. We set the position of the point-mass r to the CoM of the robot, and angular velocity $\dot{\phi}$ of the pendulum such that the velocity of the point-mass (i.e., $\dot{\phi} \times r$) equals to the normal velocity of the CoM of the robot to the lean line. Then, the desired GRF is set to the GRF of the 3D point-mass pendulum as follows:

$$f_d = -mr\left(\frac{r^T g}{r^T r} + \dot{\phi}^T \dot{\phi}\right), \tag{28}$$

where $r = PG$.[3] Given $\dot{\omega}_d$, $\ddot{\theta}_d$, and f_d, we solve Eqn. (23) to compute $\ddot{\theta}$ and f, which are then used as inputs for the inverse dynamics to determine joint torques. We use the Hybrid System dynamics algorithm [17] for this purpose.

Maintaining a Point Contact

While the robot is falling, we control the support foot so that it maintains a point contact with the ground because edge or planar contact would incur a large friction that will hinder the robot from spinning. To this end, the robot either flexes or extends the support foot if CoP is not at the corner of the foot. The foot is flexed if CoP is closer to the hill so that the CoP is located at the hill, and it is extended if CoP is closer to the toe so as to make CoP be at the toe.

SIMULATION RESULTS

We tested the proposed controller with a full-sized humanoid robot model of which height and weight are about 120 cm and 50 kg. In Fig. 6, the robot is given an external push of 160N applied to the left for 300 msec. The push makes the robot fall sideways (Fig. 6, top) when no fall control is engaged. The bottom row of Fig. 6 shows the result when the fall controller uses only the spinning strategy, without activating the bending strategy. The robot lifts and rotates the right leg, which creates a noticeable spin of the trunk as can be seen compared with no-control case (Fig. 6, top). However, the robot fails to touchdown with its backpack and its hand first collides with the ground as shown in Fig. 6(g).

Fig. 7 shows the result of using both the spinning and bending strategies under the same push as in Fig. 6. By lifting its swing leg, the robot tries to rotate the trunk counter-clockwise about the spin axis and increases the possibility to touchdown with the backpack by rotating the trunk backward. As a result, it successfully touches down with the backpack (Fig. 7(g)).

Fig. 8 shows the trajectories of spin velocity and the angle between the normal vector of the trunk (n in Fig. 5) and the downward normal direction of the ground for the experiment shown in Fig. 7. In our experiment, fall controller is triggered when the angle between the lean line and the ground normal direction exceeds 15 degrees, which is detected about 30 msec after the push. The duration for which the fall controller is active is about 400 msec and the planning of the fall is executed at every 50 msec. One can see that the actual spin velocity is progressively approaching the desired spin velocity, but it cannot achieve the desired value because it is not feasible, as can be seen by the discrepancy between the admissible and the desired spin velocity. As a result, the trunk normal vector forms about 55 degrees with the downward normal direction of the ground at the touchdown (Fig. 8, bottom). Still with this angle, the robot successfully touches down with the backpack because of the 3D profile of the backpack, which is protruded backward moderately.

In our simulation experiments, we could make the robot fall on the backpack up to 70 degrees of the push angle (θ in Fig. 9(a)). Fig. 9 shows the configuration of the robot when it touches down with the ground. The swing leg is lifted more when the push angle is 70 degrees (Fig. 9(b)) because it requires more spinning than when it is pushed 110 degrees (Fig. 9(c)).

[3] Equation 28 can be derived by the relations of a 3D point-mass pendulum:

$$m\ddot{r} = f + mg \tag{29}$$

$$\dot{r} = \dot{\phi} \times r \tag{30}$$

$$f \times r = 0, \tag{31}$$

where $r = \overrightarrow{PG}$, f is the GRF, and $\dot{\phi}$ is the angular velocity. Equation 31 states that the GRF must be collinear with $\overrightarrow{PG}$.

Copyright © 2011 by ASME

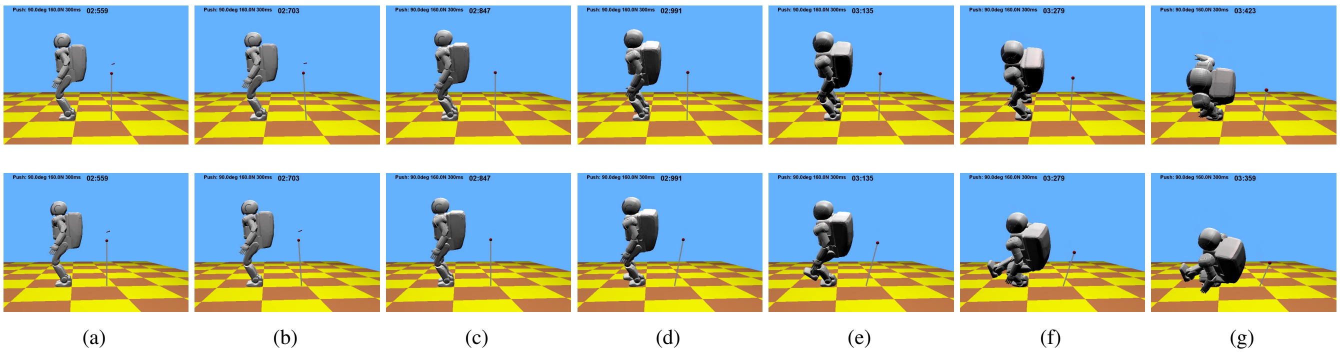


FIGURE 6. An external force of 160N applied at the CoM of the robot to its left for 300 msec makes the robot fall. Top: The robot locks all the joints without triggering the fall controller. It falls sideways. Bottom: The robot engages only the spinning strategy. It lifts and rotate its right leg to control the spin angle of the trunk. Bending strategy is not used. The trunk spins, but not enough to fall on the backpack. As a result, the robot's hand hits the ground first.

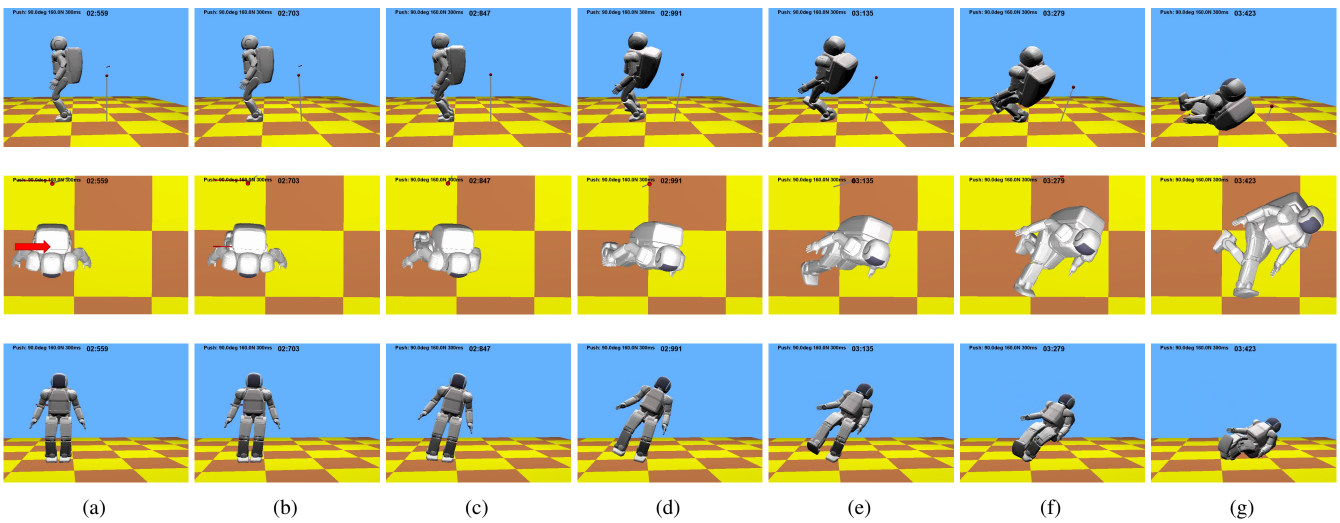


FIGURE 7. Under the same external force as in Fig. 6, the robot uses both the spinning and bending strategies. It successfully touches the ground with the backpack. Top, middle, and bottom rows show the side, top, and front views of the simulation, respectively.

DISCUSSION AND FUTURE WORK

We showed that a humanoid robot can fall on a target body segment by utilizing the dynamic coupling of the body parts. The fall control technique proposed in this paper can make the robot fall on the backpack even if it initially falls sideways. However, there is a limit in the range of falling directions that our fall control is effective. For instance, the robot cannot touchdown with the backpack if it falls forward.

The effective range of the fall control is determined by many factors such as the mass distribution and joint configuration of a robot. The range will greatly increase if a robot has a waist joint that allows for the vertical rotation of the upper body.

One possible remedy to deal with the limitation on the effective range is to attach shock-absorbing materials to multiple locations on the robot. For example, if the robot has cushioning pads on the knees, the robot will be able to reduce the damage by touching down with knees first when it is falling forward. In order to have multiple target body parts, one would need to develop a controller that first selects the optimal target body part and then tries to fall with that body part.

There also can be a different method of controlling the target body segment to collide with the ground than what is introduced here. For example, changing the foot-ground contact configuration can be effective for this purpose. If a robot can step while falling, the robot can change the falling direction and it may also be possible to change the orientation of the trunk as observed in [1].

In this paper, we made a simplifying assumption that the robot is not surrounded by objects that can possibly collide with the robot. In the real situation where a falling robot can collide

Copyright © 2011 by ASME

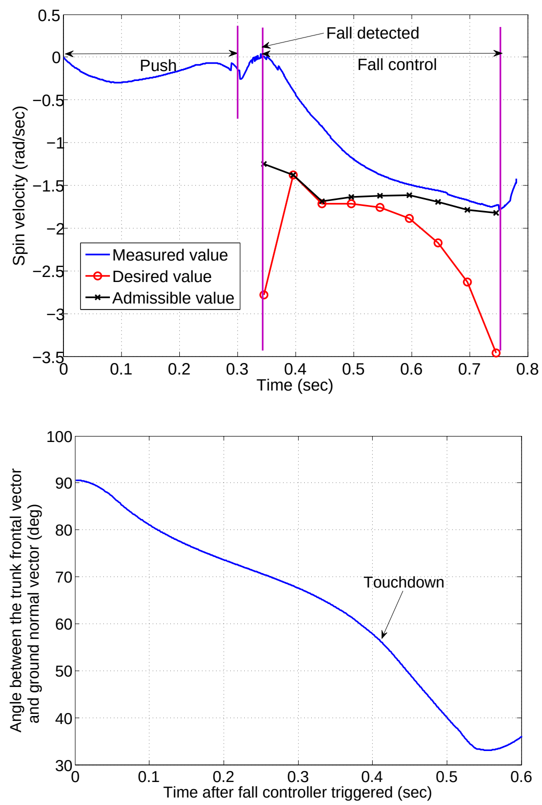
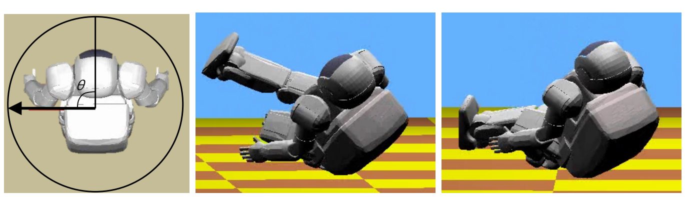


FIGURE 8. Spin velocity (top) and the angle between the normal direction of the backpack and the downward normal direction of the ground (bottom) of the fall control experiment in Fig. 7.



(a) Push angle and direction (b) 70 degrees (c) 110 degrees

FIGURE 9. Experiments with different push directions.

with surrounding objects or humans, it is important to develop an integrated approach that considers reducing the damage to the environment as well as the damage to the robot itself.

CONCLUSION

In order for humanoid robots to coexist with humans for extended time in the real world, their accidental fall must be managed so as to minimize the damage it suffers. In this paper, we propose a novel method to control a falling robot to touchdown with a targeted body part, in this case the backpack. This is achieved principally by spinning the trunk using dynamic coupling through counter-rotating the swing leg about the spin axis. This effort is helped by bending the trunk backward to increase the possibility of fall on the backpack. The assumption is that the design of the backpack is able to survive an impact much more than other parts of the robot.

ACKNOWLEDGMENT

This work was mainly done while SHL was with HRI. SHL was also supported in part by Basic Science Program of NRF funded by MEST, Korea (2010-0025725).

REFERENCES

[1] Yun, S.-K., Goswami, A., and Sakagami, Y., 2009. “Safe fall: Humanoid robot fall direction change through intelligent stepping and inertia shaping”. In IEEE Intn’l Conf. on Robotics and Automation (ICRA), pp. 781–787.
[2] Nagarajan, U., and Goswami, A., 2010. “Generalized direction changing fall control of humanoid robots among multiple objects”. In IEEE Intn’l Conf. on Robotics and Automation (ICRA), pp. 3316–3322.
[3] Fujiwara, K., Kanehiro, F., Kajita, S., Kaneko, K., Yokoi, K., and Hirukawa, H., 2002. “UKEMI: Falling motion control to minimize damage to biped humanoid robot”. In IEEE/RSJ Intn’l Conf. on Intelligent Robots and Systems (IROS), pp. 2521–2526.
[4] Fujiwara, K., F., K., Kajita, S., Yokoi, K., Saito, H., Harada, K., Kaneko, K., and Hirukawa, H., 2003, Las Vegas, NV, USA. “The first human-size humanoid that can fall over safely and stand-up again”. In IEEE/RSJ Intn’l Conf. on Intelligent Robots and Systems (IROS), pp. 1920–1926.
[5] Fujiwara, K., Kanehiro, F., Kajita, S., and Hirukawa, H., 2004, Sendai, Japan. “Safe knee landing of a human-size humanoid robot while falling forward”. In IEEE/RSJ Intn’l Conf. on Intelligent Robots and Systems (IROS), pp. 503–508.
[6] Fujiwara, K., F., K., Saito, H., Kajita, S., Harada, K., and Hirukawa, H., 2004, New Orleans, LA, USA. “Falling motion control of a humanoid robot trained by virtual supplementary tests”. In IEEE Intn’l Conf. on Robotics and Automation (ICRA), pp. 1077–1082.
[7] Fujiwara, K., Kajita, S., Harada, K., Kaneko, K., Morisawa, M., Kanehiro, F., Nakaoka, S., Harada, S., and Hirukawa, H., 2006. “Towards an optimal falling motion for a humanoid robot”. In Humanoids06, pp. 524–529.
[8] Fujiwara, K., Kajita, S., Harada, K., Kaneko, K., Morisawa, M., Kanehiro, F., Nakaoka, S., Harada, S., and Hirukawa, H., 2007. “An optimal planning of falling motions of a

Copyright © 2011 by ASME

humanoid robot”. In IEEE/RSJ Intn’l Conf. on Intelligent Robots and Systems (IROS), pp. 456–462.

[9] Ishida, T., Kuroki, Y., and Takahashi, T., 2004. “Analysis of motions of a small biped entertainment robot”. In IEEE/RSJ Intn’l Conf. on Intelligent Robots and Systems (IROS), pp. 142–147.

[10] Ogata, K., Terada, K., and Kuniyoshi, Y., Pittsburgh, 2007. “Falling motion control for humanoid robots while walking”. In Humanoids07.

[11] Ogata, K., Terada, K., and Kuniyoshi, Y., 2008, Daejeon, Korea. “Real-time selection and generation of fall damagae reduction actions for humanoid robots”. In Humanoids08, pp. 233–238.

[12] del Solar, J. R., Palma-Amestoy, R., Marchant, R., Parra-Tsunekawa, I., and Zegers, P., 2009. “Learning to fall: Designing low damage fall sequences for humanoid soccer robots”. *Robotics and Autonomous Systems,* ***57***(8), pp. 796–807.

[13] Forner Cordero, A., 2003. “Human gait, stumble and … fall?”. PhD thesis, University of Twente, Enschede, The Netherlands.

[14] Robinovitch, S. R., Hsiao, E. T., Sandler, R., Cortez, J., Liu, Q., and Paiment, G. D., 2000. “Prevention of falls and fall-related fractures through biomechanics”. *Exercise and Sports Sciences Review,* ***28***(2), April, pp. 74–79.

[15] Robinovitch, S. N., Brumer, R., and Maurer, J., 2004. “Effect of the “squat protective response” on impact velocity during backward falls”. *Journal of Biomechanics,* ***37***(9), pp. 1329–1337.

[16] Orin, D., and Goswami, A., 2008, Nice, France. “Centroidal momentum matrix of a humanoid robot: Structure and properties”. In IEEE/RSJ Intn’l Conf. on Intelligent Robots and Systems (IROS).

[17] Featherstone, R., 1987. *Robot Dynamics Algorithms*. Kluwer Academic Publishers.

Copyright © 2011 by ASME