2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Active Stabilization of a Humanoid Robot for Impact Motions with Unknown Reaction Forces

Seung-Joon Yi[*][†], Byoung-Tak Zhang[†], Dennis Hong[‡] and Daniel D. Lee[*]

Abstract— **During heavy work, humans utilize whole body motions in order to generate large forces. In extreme cases, exaggerated weight shifts are used to impart large impact forces. There have been approaches to design stable whole body impact motions based on precise dynamic models of the robot and the target object, but they have practical limitations as the uncertainty in the ensuing reaction forces can lead to instability. In the current work, we describe a motion controller for a humanoid robot that generates impacts at an end effector while keeping the robot body balanced before and after the impact. Instead of relying on the accuracy of the impact dynamics model, we use a simplified model of the robot and biomechanically motivated push recovery controllers to reactively stabilize the robot against unknown perturbations from the impact. We demonstrate our approach in physically realistic simulations, as well as experimentally on a small humanoid robot platform.**

Keywords: humanoid robot, impact motion, uncertain reaction force, biomechanically motivated push recovery

I. INTRODUCTION

One key advantage of humanoid robots is that they can seamlessly operate in human environments and work with objects designed for humans. However, object manipulation can be difficult for a humanoid robot due to its upright posture and stability issues arising from its relatively small footprint and high center of mass. Thus, a large portion of humanoid robot research has focused on realizing stable bipedal locomotion without losing balance.

There is also difficulty in exerting large forces in object manipulation since the ensuing large reaction forces can destabilize the balance of the robot. Humans instinctively modify their body posture during heavy work, and previous studies have investigated pushing heavy objects in simulation [1], [2], and on physical robots using optimized postures [3]. Full body posture control in conjunction with force sensors to lift an object with unknown mass was shown in [4].

When static forces are insufficient to accomplish a given task, humans can utilize dynamic momentum transfer by imparting an impact motion to an object. Examples of impact motions in robotics include drumming on the HRP-2 robot, which showed that impact motion of the arms can apply 50% more force than quasi-static motions to an object [5]. Other examples of impact motions using humanoid robots include a nailing task with a hammer [6], wooden plate breaking [7], [8], ball kicking [9], and dynamic lifting [10].

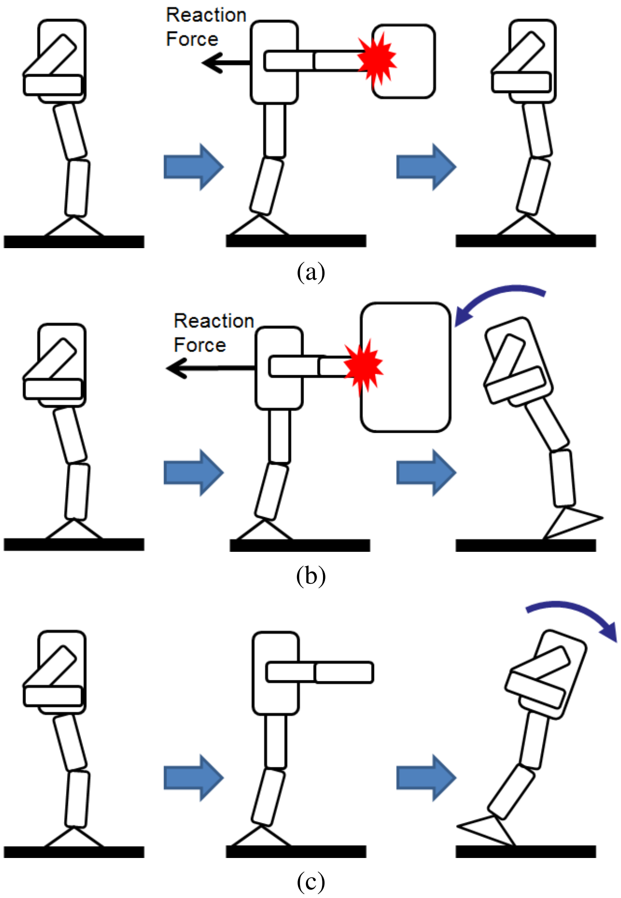


Fig. 1. Possible outcomes of a pre-designed impact motion applied to targets with different dynamics. (a) A pre-designed impact motion works well if the impact dynamics matches assumptions. (b) If the reaction force is larger than expected, the robot may fall backwards. (c) If there is less reaction force than expected, the robot may fall forwards due to excessive momentum.

In these works, the reaction forces upon impact are typically not large enough to hamper stability and were not considered explicitly. But when the impact motion becomes large, the reaction forces need to be properly accounted to ensure the post-impact stability of the robot. There have been optimzation-based approaches to generate a stable impact motion given the impact dynamics model of the robot and the object [11], but they have two practical issues for implementation on physical robots. The first one is that due to uncertainty in the reaction force, an impact motion designed for a particular task may fail when the targets display unmodeled impact dynamics. Figure 1 illustrates the

[*] GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 {yiseung,ddlee}@seas.upenn.edu
[†] BI Laboratory, Seoul National University, Seoul, Korea. btzhang@ceas.snu.ac.kr
[‡] RoMeLa Laboratory, Virginia Tech, Blacksburg, VA 24061 dhong@vt.edu

978-1-4673-1735-1/12/$31.00 ©2012 IEEE

difficulty in accommodating impact motions with unknown reaction forces. The second one is that optimization-based approaches are offline methods and cannot be utilized for the cases when motions are generated in real time, such as a teleoperation situation.

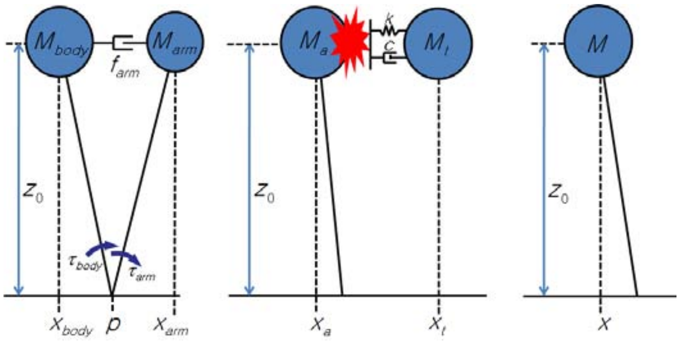
In the current work, we focus on the problem of generating a stable impact motion for a humanoid robot with unknown reaction forces from a different perspective. In contrast to previous approaches which either assume that reaction force is negligible or accurately known in advance, we regard the reaction forces as unknown disturbances applied to the robot, and use reactive push recovery controllers to stabilize the robot against them. A simple model of the robot is used to generate pre-impact motion to maximize the impulse force, while keeping the state of the robot within the stability region of the push recovery controller. We validate our approach using physically realistic simulations, as well as experimentally on the DARwin-OP small humanoid robot platform. Experimental results show that our method can successfully apply impact forces on objects with very different inertial properties without falling down.

The remainder of the paper proceeds as follows. Section II describes how impact motions are designed for a humanoid robot utilizing simple dynamic models. Section III reviews our hierarchical biomechanically motivated push recovery controller and models its stability boundaries. Section IV shows results using a physics-based simulation. Section V describes and presents experiments on the DARwin-OP humanoid robot. Finally, we conclude with a discussion of outstanding issues and potential future directions arising from this work.

II. Designing impact motions for humanoid robots

The impact motion can be divided into three phases, the pre-impact phase, the impact phase and the post-impact phase. At the pre-impact phase, the robot accelerates its end effector so that it hits the target with large momentum. During the impact phase, the end effector hits the target and momentum is transferred between the robot and target. In the post-impact phase, the robot stabilizes itself from the ensuing perturbation due to reaction forces imparted during the impact. We describe each phase in more detail in this section.

A. Pre-impact phase



(a) $t < T_1$ (b) $t = T_1$ (c) $t > T_1$

Fig. 2. Simplified models for the three different phases of the impact motion. (a) Pre-impact model with torso mass M_{body} and arm mass M_{arm}. (b) Impact model with robot and target virtual masses M_a and M_t. (c) Post-impact model with single point mass with $M = M_{body} + M_{arm}$.

At the pre-impact phase, the robot needs to accelerate its end effector as much as possible to build up linear momentum, while being stable. To model the dynamics of the robot in this phase, we use an extended linear inverted pendulum model (LIPM) with a secondary mass as in Figure 2 (a). This model has center of mass (COM) height z_0, torso mass M_{body}, arm mass M_{arm}, and horizontal COM positions of torso and arm from support point are denoted by x_{body} and x_{arm}. As we assume LIPM for each mass, the torques at the point p due to each mass are

$$\tau_{body} = M_{body} z_0 \left(\omega^2 (x_{body} - p) - \ddot{x}_{body}\right) \quad (1)$$

$$\tau_{arm} = M_{arm} z_0 \left(\omega^2 (x_{arm} - p) - \ddot{x}_{arm}\right), \quad (2)$$

where $\omega = \sqrt{\frac{g}{z_0}}$ and g is the gravitational constant. If we denote the total robot mass as $M = M_{body} + M_{arm}$, then they should satisfy following equation to make net moment zero at point p

$$p = \frac{M_{body}(\omega^2 x_{body} - \ddot{x}_{body}) + M_{arm}(\omega^2 x_{arm} - \ddot{x}_{arm})}{M\omega^2}, \quad (3)$$

which ensures the dynamic stability of the robot if the position of p lies inside the support polygon during the motion.

To generate the motion trajectories, we can model the arm as a linear joint

$$f_{arm} = x_{arm} - x_{body}, \quad (4)$$

and simultaneously optimize two variables x_{body} and f_{arm} to maximize the end effector velocity at impact with the constraints on $\dot{f}_{arm}$ and $\ddot{f}_{arm}$ to regulate the maximum velocity and the force of the joint. However as it is hard to approximate the joint constraints for robots with rotary joints, we take a simpler approach of designing the arm motion first and use it to update the torso trajectory using (3)

$$\ddot{x}_{body} = \omega^2 (x_{body} - p) + M_{arm}(\omega^2 f_{arm} - \ddot{f}_{arm})/M, \quad (5)$$

with initial condition $x_{body}(0) = x_0$ and $\dot{x}_{body}(0) = 0$.

B. Impact phase

At the impact phase, the end effector hits the target, making a momentum transfer between the robot and the target object. One way to model this is using a mass-spring-damper model with two virtual masses M_a and M_t [12] as shown in Figure 2 (b), which is also affected by the posture of the robot. However as we consider the case when the reaction force is not precisely known in advance, we use following simplified impact dynamics model, which assumes that the arm and the torso forms a single rigid body after instantaneous impact

$$x(T_1) = (M_{body} x_{body}(T_1) + M_{arm} x_{arm}(T_1))/M \quad (6)$$

$$\dot{x}(T_1) = (M_{body} \dot{x}_{body}(T_1) + M_{arm} \dot{x}_{arm}(T_1) + P_{impact})/M, \quad (7)$$

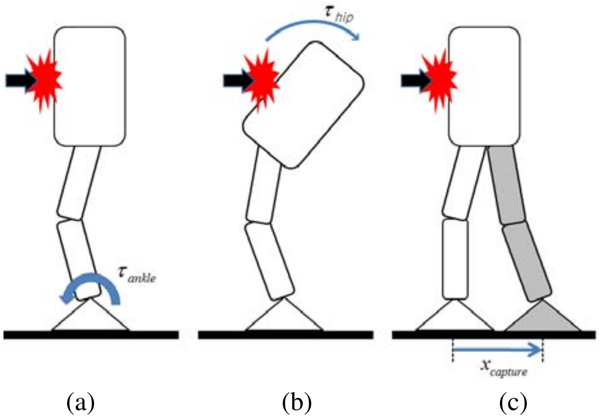


Fig. 3. Three biomechanically motivated push recovery strategies. (a) ankle strategy (b) hip strategy (c) step strategy

where T_1 is the time of impact, x is the horizontal COM position for the LIPM right after impact, $P_{impact} \leq 0$ is the instantaneous momentum change due to impact.

C. Post-impact phase

After the impact phase is over, the robot should be able to stabilize itself. We assume the single mass model in Figure 2 (c) for $t > T_1$ with the initial state $(x(T_1), \dot{x}(T_1))$ from (6), (7) to straightforwardly apply the LIPM based push recovery controllers. We should design the torso and arm trajectory so that the resulting state $(x(T_1), \dot{x}(T_1))$ right after impact lies in the stability region of the push recovery controller for a broad range of P_{impact}. We cover more detail about the push recovery controllers and their stability regions in next section.

III. Full-Body Push Recovery Controllers

Biomechanical studies show that humans display three distinctive motion patterns in response to sudden external perturbations, which we denote as ankle, hip and step push recovery strategies [13] and are shown in Figure 3. In this section we review three push recovery controllers based on those strategies using a simplified model of the robot, and provide how they can be selected based on current state and the stability region of each controller.

A. Ankle push recovery

The ankle strategy applies control torque on the ankle joints to keep the center of mass within the base of support. We can assume the abstract model in Figure 4 (a), where ankle torque τ_{ankle} is applied to a LIPM with mass M, COM height z_0 and COM horizontal position x from current support point. Then the resulting linearized dynamic model is

$$\ddot{x} = \omega^2(x - \tau_{ankle}/Mg), \tag{8}$$

which can be controlled by a PD-control on x

$$\tau_{ankle} = K_p x + K_d \dot{x}, \tag{9}$$

where K_p and K_d are the control gains.

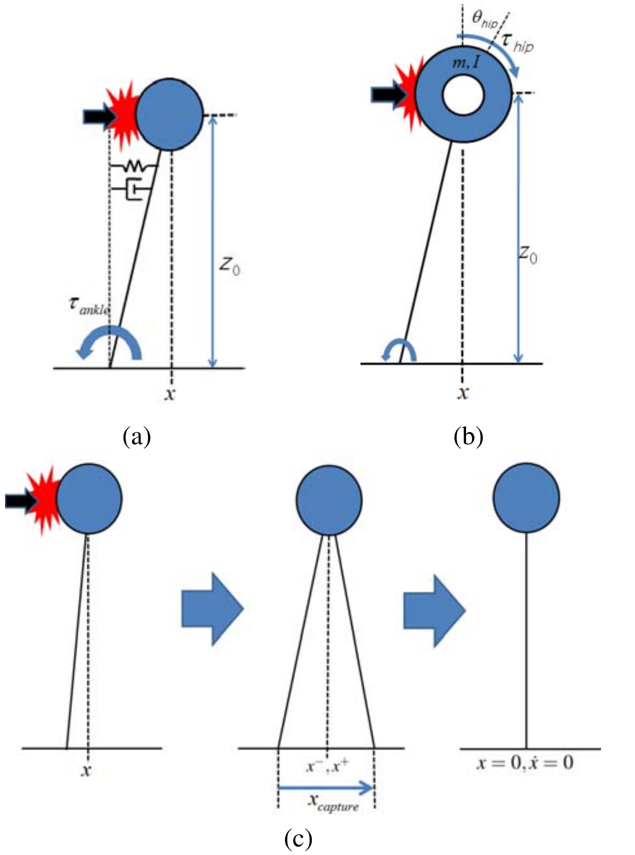


Fig. 4. Three push recovery strategies. (a) Ankle strategy that applies control torque at the ankle joint. (b) Hip strategy which uses angular acceleration of torso and limbs to apply counteractive ground reaction forces. (c) Step strategy that changes the support point by stepping.

B. Hip push recovery

The hip strategy uses angular acceleration of the torso and limbs to generate a backward ground reaction force (GRF) to pull the center of mass back towards the base of support. The abstract model in Figure 4 (b) includes a flywheel with point mass at height z_0 and rotational inertia I, and control torque τ_{hip} at the COM. Then the resulting linearized dynamic model is

$$\ddot{x} = \omega^2(x - \tau_{hip}/Mg) \tag{10}$$

$$\ddot{\theta}_{hip} = \tau_{hip}/I. \tag{11}$$

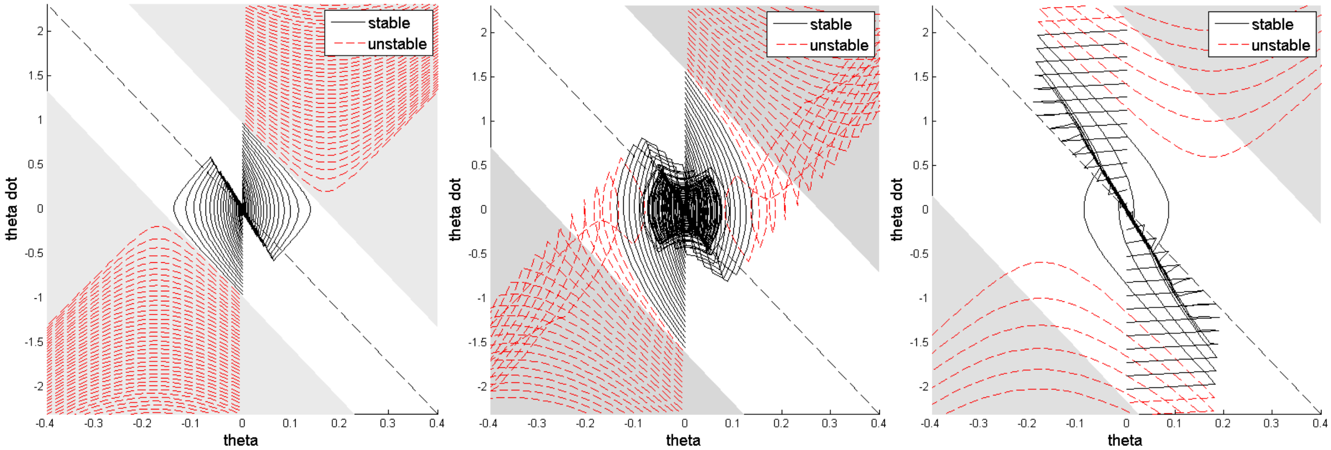
However we should stop the flywheel from exceeding joint limits. In this case, following bang-bang profile [14] can be used for applying hip torque to maximize the effect while satisfying the joint angle constraint

$$\tau_{hip}(t) = \begin{cases} \tau_{hip}^{MAX} & 0 \leq t < T_{H1} \\ -\tau_{hip}^{MAX} & T_{H1} \leq t < 2T_{H1}, \end{cases} \tag{12}$$

where τ_{hip}^{MAX} is the maximum torque that the can be applied on torso and T_{H1} is the time the torso stops accelerating.

C. Step push recovery

The step strategy moves the base of support towards the direction of push by taking a step, as shown in Figure 4 (c). If we assume the support point transition occurs instantly



(a) Ankle strategy (b) Hip strategy (c) Step strategy

Fig. 5. Stability regions for each push recovery controller. White and gray region denotes stable and unstable region of state space. Black and red lines denote stable and unstable state trajectories from various initial states.

preserving the linear momentum, we can get following landing position from initial support point [14].

$$x_{capture} = \dot{x}/\omega + x. \tag{13}$$

D. High-level push recovery controller

When pushed, humans perform a combination of push recovery behaviors according to the particular situation. To select the appropriate set of push recovery behaviors as humans do, we use a hierarchical controller where ankle, hip and step push recovery controllers work as low-level subcontrollers and the high-level push recovery controller triggers each according to the current sensory input [15].

For the simplified models shown in Figure 4, previous analysis have shown the decision boundaries of each controller based on the current state [16]. If we assume maximum ankle torque as τ^{ankle}_{MAX}, then the stability region for ankle push recovery controller is derived as

$$|Mg(\dot{x}/\omega + x)| < \tau^{MAX}_{ankle} \tag{14}$$

which is increased by combining the hip strategy plus ankle strategy

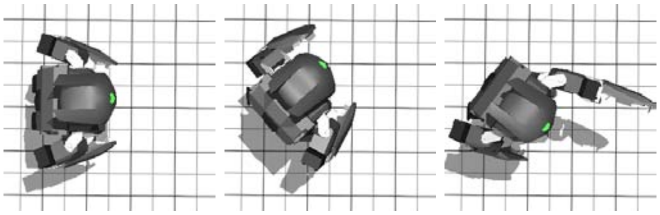
$$|Mg(\dot{x}/\omega + x)| < \tau^{MAX}_{ankle} + \tau^{MAX}_{hip}(e^{\omega T_{H1}} - 1)^2. \tag{15}$$

Finally, if we assume instantaneous support point transition without loss of linear momentum, we have the following stability region for using all three strategies at once:

$$|Mg(\dot{x}/\omega + x)| < \tau^{MAX}_{ankle} + \tau^{MAX}_{hip}(e^{\omega T_{H1}} - 1)^2 + Mgx^{MAX}_{capture}, \tag{16}$$

where $x^{MAX}_{capture}$ is the maximum step size available. In this case we can use two boundary conditions in (14) and (15) to select between controllers based on current state. Phase space trajectory plots and stability regions for each controller are shown in Figure 5.

For the more realistic case with a multi-segmented body with motor dynamics as on a physical robot, these theoretical boundaries do not fit well and the high-level controller needs to be trained from experience [15], [17]. We do not cover the learning algorithm here in detail due to lack of space.



(a) (b) (c)

Fig. 6. Three upper body keyframes for generating the end effector movement,

IV. Simulation Results

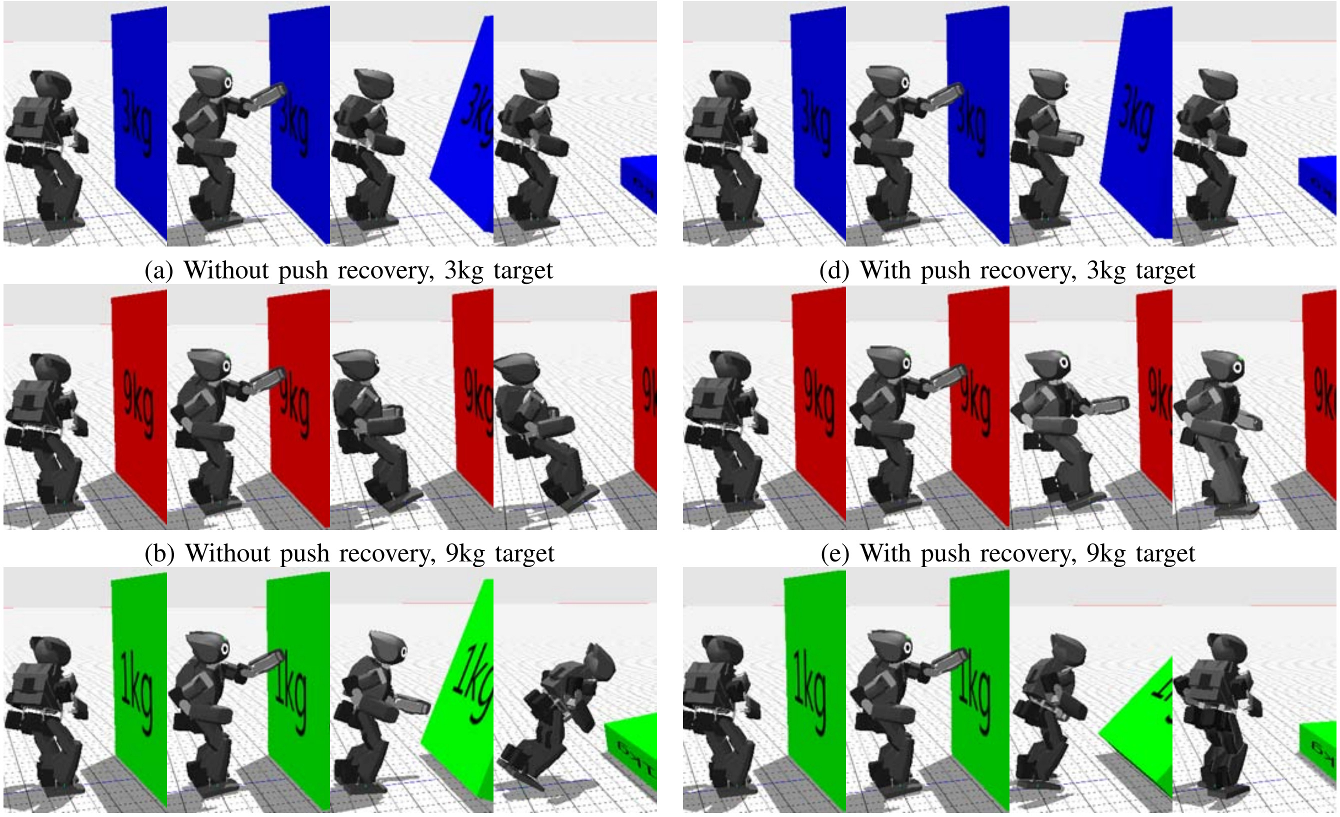
A. Simulation setup

We use the commercial Webots robot simulator [18] based on the Open Dynamics Engine(ODE) physics library with the supplied simulated model of the DARwIn-OP commercial humanoid robot. The DARwIn-OP robot is 45cm tall, weighs 2.8kg, and has 3-axis accelerometer and 3-axis gyroscope for inertial sensing. Our impact motion controller with push recovery is implemented using our modular open source humanoid framework [19]. The controller update frequency and physics simulation frequency are both set to 100 Hz.

We consider the situation where a humanoid robot knocks another object down by punching. For the target, we use a uniformly dense rectangular solid with the same COM height and support base length as DARwIn-OP robot, which is set in an upright position 30 cm in front of the robot.

B. Motion generation

As the DARwIn-OP robot has wide shoulders and 3 degree of freedom arms, body rotation is necessary to design a punch motion that can hit an object directly in front of the robot. We interpolate three upper body keyframes shown in Figure 6 to generate the arm motion f_{arm}. Torso trajectory is generated using (5) with parameters $x_0 = -0.04$, $M = 2$, $m = 0.2$ and $p = -0.15$, where p is found by repeated trials against 3kg target. For stance parameters we use COM height $z_0 = 0.295$, ankle width $d_{stance} = 0.75$ and body frontal tilt



(a) Without push recovery, 3kg target

(d) With push recovery, 3kg target

(b) Without push recovery, 9kg target

(e) With push recovery, 9kg target

(c) Without push recovery, 1kg target

(f) With push recovery, 1kg target

Fig. 7. Result of applying impact motions with and without push recovery control to targets with different masses in simulated environment.

angle $\theta_{torso} = 20°$. The push recovery controller is triggered 0.3s after movement starts. We use the ankle and the step strategy for push recovery, and decision boundaries of (14) with heuristic parameters are used to select between the ankle and the step strategy based on current state estimated from inertial sensor readings. For push recovery controller parameters, we use values of $K_p'' = 0$, $K_d'' = 0.15$, $x_{capture}^{MAX} = 0.06$ and step duration t_{STEP} = 0.35.

C. Results

Figure 7 summarizes the result of applying the impact motion controller against targets with 1kg, 3kg and 9kg masses, with and without push recovery. We can see that the pre-designed impact motion does not work well if the impact dynamics are different from the initial assumptions. On the other hand, our approach can stabilize the robot using ankle torque and reactive stepping against a wider range of perturbations from the impact.

V. Experimental Results

A. Hardware setup

We use a physical DARwIn-OP robot to validate our approach experimentally. The DARwIn-OP robot has position-controlled Dynamixel servos for actuators, which are controlled by a custom microcontroller connected to an Intel Atom-based embedded PC at a control frequency of 100Hz. Instead of using targets with different masses, we used a single target composed of a cardboard box with 2.8kg of weight attached at the bottom, and changed the distance between the robot and target.

B. Motion generation

With help of our modular open source humanoid framework, the same controller code is used with different I/O libraries. We used the same parameters for motion generation, except for slightly different upper body keyframe to take the non-ideal servo dynamics into account.

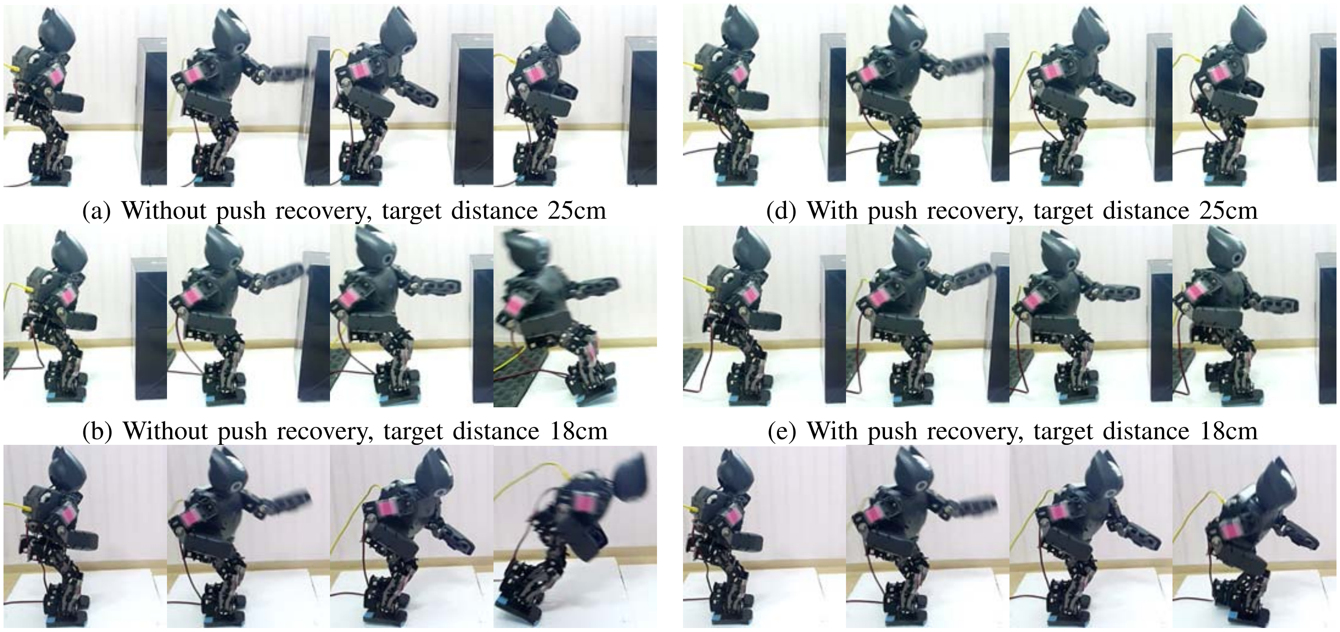
C. Results

Figure 8 shows the result of applying the impact motion controller for DARwIn-OP robot against targets with different distances[1]. We can see that our approach can help the DARwIn-OP humanoid robot to stabilize itself against unknown perturbations from the impact.

VI. Conclusions

In this work, we describe a motion controller for a humanoid robot that can generate impulsive impacts at its end effector while keeping the robot in balance before and after the impact. Instead of relying on the precise model of the robot and prior knowledge about the reaction force, we view the reaction forces as unknown perturbations and use biomechanically motivated push recovery controllers to

[1] `http://youtu.be/KIStPiGaxxI`



(a) Without push recovery, target distance 25cm

(d) With push recovery, target distance 25cm

(b) Without push recovery, target distance 18cm

(e) With push recovery, target distance 18cm

(c) Without push recovery, no target

(f) With push recovery, no target

Fig. 8. Result of applying impact motion using a DARwIn-OP robot with and without push recovery control to targets with different distances.

reactively stabilize the robot, which also enabled us to use a simplified model of the robot for motion generation. Our approach is implemented and demonstrated in physically realistic simulations and experimentally on a DARwIn-OP small humanoid robot. The experimental results show that our methods can effectively stabilize the robot from unknown perturbations across a variety of impact forces, and another benefit of our approach is that it can also be used when motions are generate in real time as it does not rely on prior knowledge of reaction force.

Possible future work includes extending current approach to real-time teleoperation control, incorporating learning algorithms to learn impact dynamics with physical robots, and implementation of these algorithms on full-sized humanoid robots.

Acknowledgments

We acknowledge the support of the NSF PIRE program under contract OISE-0730206 and ONR SAFFIR program under contract N00014-11-1-0074. This work was also partially supported by the NRF grant of MEST (0421-20110032), the IT R&D program of MKE/KEIT (KI002138, MARS), and the ISTD program of MKE (10035348).

References

[1] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, “Pushing manipulation by humanoid considering two-kinds of zmps,” in *ICRA*, 2003, pp. 1627–1632.

[2] Y. Hwang, A. Konno, and M. Uchiyama, “Whole body cooperative tasks and static stability evaluations for a humanoid robot,” in *IROS*, 2003, pp. 1901–1906.

[3] A. Konno, Y. Hwang, S. Tamada, and M. Uchiyama, “Working postures for humanoids robots to generate large manipulation force,” in *IROS*, 2005, pp. 1788–1793.

[4] K. Harada, S. Kajita, H. Saito, M. Morisawa, F. Kanehiro, K. Fujiwara, K. Kaneko, and H. Hirukawa, “A humanoid robot carrying a heavy object.” in *ICRA*, 2005, pp. 1712–1717.

[5] A. Konno, T. Matsumoto, Y. Ishida, D. Sato, and M. Uchiyama, “Drum beating and a martial art bojutsu performed by a humanoid robot,” *Humanoid Robots: New Developments*, 2007.

[6] T. Tsujita, A. Konno, S. Komizunai, Y. Nomura, T. Owa, T. Myojin, Y. Ayaz, and M. Uchiyama, “Analysis of nailing task motion for a humanoid robot,” in *IROS*, 2008, pp. 1570–1575.

[7] T. Matsumoto, A. Konno, L. Gou, and M. Uchiyama, “A humanoid robot that breaks wooden boards applying impulsive force.” in *IROS*, 2006, pp. 5919–5924.

[8] A. Konno, T. Myojin, T. Matsumoto, T. Tsujita, and M. Uchiyama, “An impact dynamics model and sequential optimization to generate impact motions for a humanoid robot,” *International Journal of Robotic Research*, vol. 30, pp. 1596–1608, Nov. 2011.

[9] J. Müller, T. Laue, and T. Röfer, “Kicking a ball - modeling complex dynamic motions for humanoid robots,” in *RoboCup 2010: Robot Soccer World Cup XIV*, vol. 6556, 2011, pp. 109–120.

[10] H. Arisumi, S. Miossec, and J.-R. C. abd K. Yokoi, “Dynamic lifting by whole body motion of humanoid robots,” in *IROS*, 2008, pp. 668 – 675.

[11] T. Tsujita, A. Konno, and M. Uchiyama, “Optimization of impact motions for humanoid robots considering multibody dynamics and stability,” in *IROS*, 2010, pp. 718–725.

[12] H. Asada and K. Ogawa, “On the dynamic analysis of a manipulator and its end effector interacting with the environment,” in *ICRA*, vol. 4, 1987, pp. 751–756.

[13] A. G. Hofmann, “Robust execution of bipedal walking tasks from biomechanical principles,” Ph.D. dissertation, Cambridge, MA, USA, 2006.

[14] J. Pratt, J. Carff, and S. Drakunov, “Capture point: A step toward humanoid push recovery,” in *in 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[15] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, “Learning full body push recovery control for small humanoid robots.” in *ICRA*, 2011.

[16] B. Stephens, “Humanoid push recovery,” in *Proceedings of the IEEE RAS International Conference on Humanoid Robots*, 2007.

[17] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, “Online learning of a full body push recovery controller for omnidirectional walking,” in *Proceedings of the IEEE RAS International Conference on Humanoid Robots*, 2011, pp. 1–6.

[18] Webots, “http://www.cyberbotics.com,” commercial Mobile Robot Simulation Software. [Online]. Available: http://www.cyberbotics.com

[19] S. G. McGill, J. Brindza, S.-J. Yi, and D. D. Lee, “Unified humanoid robotics software platform,” in *The 5th Workshop on Humanoid Soccer Robots*, 2010.