# Trajectory-Free Reactive Stepping of Humanoid Robots Using Momentum Control

Hyunchul Choi[1], Sukwon Lee[2], Tail Jin[3], and Sung-Hee Lee[4]

***Abstract*— Momentum control approaches have been successfully applied to balance controllers for humanoid robots due to their integral controllability of ground reaction force and the center of pressure. In this paper, we introduce a trajectoryfree reactive stepping controller using momentum control. The controller is characterized by moving passively in the direction of external pushes without attempting to follow some prescribed trajectory, thereby achieving a natural reactive stepping behavior adaptive to various perturbations. Constructed on top of momentum control, our stepping controller is realized straightforwardly by setting a set of suitable inputs such as the desired momentum rate change and the target swing foot position to the momentum controller for each phase of stepping. We demonstrate the effectiveness of our approach through various simulation experiments on a humanoid robot model.**

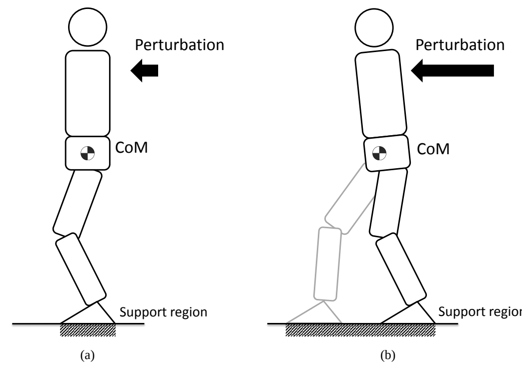



Fig. 1. Balance recovery strategies: (a) Postural balance strategy and (b) reactive stepping strategy.

## I. INTRODUCTION

Momentum-based control approach that controls both the linear and angular momenta of a humanoid robot has been studied steadily in robotics [1]–[5]. Since the ground reaction force (GRF) and the center of pressure (CoP) of a robot, most important features regarding humanoid balance, are uniquely determined by the rate of change of linear and angular momenta, momentum control approaches have shown to be effective for balance maintenance [6].

Kajita et al. [1] introduced angular momentum control for the whole body control problem. Abdallah and Goswami [2] developed a postural balance controller that controls the rate of change of linear and angular momenta of a humanoid robot. More recently, Micchietto et al. [4] proposed a whole body postural balance controller that identifies the desired CoP as the high level input. Lee and Goswami [7] extended the method of [4] to present a balance controller for a non-level and non-stationary ground.

Among a number of strategies that a humanoid robot can employ to maintain balance, two most representative strategies are the postural balance strategy and the reactive stepping strategy as shown in Fig. 1. The postural balance strategy is usually chosen for relatively short and mild perturbations, against which a humanoid robot can maintain balance in place simply by controlling the ankle or hip joints [8], [9], or by rotating the whole upper body [2]–[5], [7]. Upon the long or strong perturbations, a humanoid robot should take a reactive stepping strategy in which a robot takes one or more steps to prevent falling [10]–[13].

Recently, researchers have applied momentum control approaches to reactive stepping. Wu and Zordan [14] introduced a momentum-based stepping controller that generates parameterized curves for the swing foot and center of mass (CoM) trajectories according to the step position and duration. The method subsequently creates whole body motions to realize the trajectories via joint accelerations optimally calculated from the multiobjective function and joint torques computed by inverse dynamics. Yun and Goswami [11] presented a novel reactive stepping method based on a rimless wheel model. Specifically, they developed the *generalized foot placement estimator* (GFPE) to define the target stepping point. The proposed method is applicable to non-level ground.

In both momentum-based stepping studies discussed above, the stepping motion is created from the planned trajectories of CoM and the swing foot. Such trajectories are usually represented as splines or some simple polynomial curves, and scaled to satisfy the boundary conditions such as the starting and target configurations. However, such trajectory-based reactive stepping approaches have some limitations. 1) In many cases, the desired trajectories generate nicely smooth geometric curves but its implication on dynamics is lacking. 2) For that reason, it is typically difficult to find suitable control parameters to follow the desired trajectories. For instance, if a PD-controller is used for following the trajectory, excessive gains will create wobbling behavior around the target trajectory. Parameter tuning is particularly difficult for the reactive stepping controller because the controller must work under various perturbation

[1]H. Choi is with LG Electronics, Seoul, Korea hcchoi@gist.ac.kr
[2]S. Lee is with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea sukwonlee@kaist.ac.kr
[3]T. Jin is with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea jin219219@kaist.ac.kr
[4]S.-H. Lee is with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea sunghee.lee@kaist.ac.kr

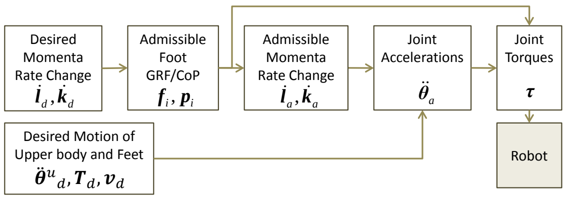



Fig. 2. Momentum control module [7].

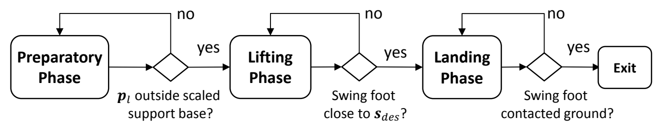



Fig. 3. Phase diagram of reactive stepping control.

conditions. 3) If an additional perturbation is applied during the reactive stepping, the desired trajectories may need to be updated, which requires non-trivial efforts to switch to the new trajectories.

In this paper, we introduce a trajectory-free reactive stepping controller using momentum control. The controller is characterized by moving passively in the direction of the perturbation without attempting to follow the desired trajectory under a perturbation. The yielding to the perturbed direction leads to a natural reactive stepping behavior. Of course, the concept of utilizing passiveness for walking is not new and has been investigated extensively in many literatures (e.g., [15], [16]). In this work, we show that a reactive stepping controller can be realized in a very straightforward manner under momentum control framework, without requiring planned trajectories for CoM and the swing foot. Specifically, our reactive stepping controller consists of three phases: In the preparatory phase, the robot identifies the swing and support feet and shifts CoM on the horizontal plane toward the support foot while passively moving in the pushed direction. Next, the robot lifts the swing foot and lands it on the target stepping point. On a strong push, the controller can take multiple steps to keep balance. Our approach has the advantages that it does not require a trajectory generator and the corresponding controller to follow the trajectory, and that the controller can create adaptive stepping motion for various perturbations.

## II. Preliminary: Momentum Controller

Given the desired momentum rate change, one can compute the joint torques to realize the momentum rate change as closely as possible. A detailed procedure can be found in [7], and here we present a brief summary: First, given the desired linear and angular momentum rate changes (which are not necessarily physically realizable by a robot), the admissible (physically realizable) foot GRFs and foot CoPs are determined such that they can create admissible momentum rate change that is as close as possible to its desired value. Subsequently, the joint accelerations are determined so as to create the admissible momentum rate change. As there are infinitely many solutions for the joint accelerations given the admissible momentum rate change, we can impose the preferred motion for the upper body and the swing foot as the secondary objective, so that the resulting joint accelerations will satisfy the given momentum rate change while generating the desired motion as close as possible. Finally, inverse dynamics is performed to compute the necessary joint torques to create the joint accelerations. Fig. 2 shows the framework of the momentum control.

If the desired linear and angular momentum rate changes are both admissible, the final joint torques can realize the desired values. Otherwise, one should choose either linear or angular momentum to satisfy and sacrifice the other. If linear momentum is chosen to be respected, the controller sacrifices the angular momentum: The humanoid robot rotates the upper body to the direction of the perturbation to maintain balance in place, which corresponds to the postural balance behavior. On the other hand, if angular momentum is chosen to be satisfied, the robot maintains the upper body upright and allows for the linear momentum to be generated in the direction of the perturbation. This strategy requires reactive stepping motion lest it should not topple.

In summary, the behavior of the momentum controller is determined by the set of inputs, the desired linear and angular momentum rate change ($\dot{\boldsymbol{l}}_d$ and $\dot{\boldsymbol{k}}_d$), the desired joint accelerations for the upper body ($\ddot{\boldsymbol{\theta}}_d^u$), and the desired position and velocity of the swing foot ($\boldsymbol{T}_d, \boldsymbol{v}_d$).

For example, in case of postural balance control, $\dot{\boldsymbol{l}}_d$ and $\dot{\boldsymbol{k}}_d$ can be determined as

$$\dot{\boldsymbol{l}}_d = \boldsymbol{\Gamma}_{lp} m(\boldsymbol{r}_{Gd} - \boldsymbol{r}_G) + \boldsymbol{\Gamma}_{ld}(\boldsymbol{l}_d - \boldsymbol{l}) \quad (1)$$

$$\dot{\boldsymbol{k}}_d = \boldsymbol{\Gamma}_{kd}(\boldsymbol{k}_d - \boldsymbol{k}) \quad (2)$$

where $m$ is the total mass of the robot, $\boldsymbol{\Gamma}_{lp}$ and $\boldsymbol{\Gamma}_{ld}$ are proportional and derivative gains, $\boldsymbol{r}_G$ is the location of CoM, and $\boldsymbol{l}$ and $\boldsymbol{k}$ are linear and angular momenta, respectively. Subscript $_d$ denotes the desired value of each property. Postural balance can be maintained if we set $\boldsymbol{l}_d = 0$ and $\boldsymbol{k}_d = 0$ to stabilize the momentum generated by the perturbation. $\boldsymbol{r}_{G,d}$ can be set such that the ground projection of CoM is located in the middle of the two feet.

## III. Trajectory-Free Reactive Stepping Strategy

Our reactive stepping control is constructed based on the momentum controller. As Fig. 3 shows, it consists of thres phases: Preparatory, lifting, and landing phases. In preparatory phase, the robot shifts its CoM laterally toward the support foot to prevent toppling during the foot swing. At the same time, it moves passively to the pushed direction. In lifting phase, the robot lifts the swing foot to some target position, and subsequently in landing phase, the robot attempts to land the swing foot to the target stepping point. Each phase operates sequentially: when a certain condition is satisfied in a phase, the controller makes transition to the next phase. After the landing phase, the higher level controller checks the stability and determines whether to resume the reactive stepping controller for additional stepping or trigger

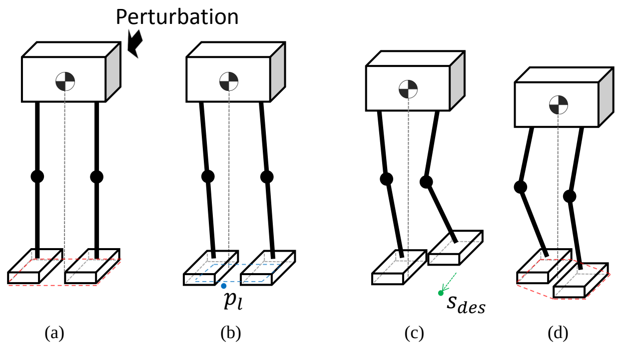



Fig. 4. (a) When a robot is pushed, the high level controller determine whether to trigger the reactive stepping controller or not. (b) In preparatory phase, the robot attempts to move its CoM to the support foot while passively moving to the pushed direction. If an indicator $p_l$ leaves the scaled support polygon, the controller proceeds to lifting phase. (c) In lifting phase, the robot lifts the swing foot to some desired position, which is determined from the desired stepping point $s_{des}$. The controller updates $s_{des}$ repeatedly at every control time step during the lifting phase to deal with continuous perturbations. (d) In landing phase, the robot attempts to land the swing foot to $s_{des}$.

a postural balance controller. The configurations of a humanoid robot's legs for each phase are illustrated in Fig. 4.

The reactive stepping controller is constructed on top of the momentum controller. Hence, the desired control action is achieved by giving suitable inputs to the momentum controller. Similarly to the postural balance control, we use (1) and (2) in order to determine the desired momentum rate changes $\dot{\boldsymbol{l}}_d$ and $\dot{\boldsymbol{k}}_d$. Therefore, the total inputs to the momentum controller are $\boldsymbol{r}_{Gd}, \boldsymbol{l}_d, \boldsymbol{k}_d, \ddot{\boldsymbol{\theta}}_d^u, \boldsymbol{T}_d$, and $\boldsymbol{v}_d$. Throughout the stepping control, $\boldsymbol{k}_d$ is set to zero in order to stabilize the rotational motion. $\ddot{\boldsymbol{\theta}}_d^u$ is determined such that the upper body maintains the default pose. Other inputs are determined differently per each phase as detailed next.

### A. Preparatory Phase

When the reactive stepping controller is triggered, the controller first enters the preparatory phase, in which the robot identifies the swing and support feet and shifts CoM laterally toward the support foot. Meanwhile, the controller accepts the current linear momentum (created by the perturbing external force) and moves passively in that direction. We achieve this behavior by setting the inputs as

$$\boldsymbol{l}_d = \boldsymbol{l} \tag{3}$$

$$\boldsymbol{r}_{Gd} = \boldsymbol{r}^*_{Gd} + \alpha \boldsymbol{s}_{lat} \tag{4}$$

Equation (3) ensures that we accept the current linear momentum as the desired value, and do not attempt to drive the linear momentum to some other value. Therefore, $\dot{\boldsymbol{l}}_d$ is determined solely from $\boldsymbol{r}_{Gd}$, which is set by (4). $\boldsymbol{r}^*_{Gd}$ denotes the position of which ground projection is equal to the center of the two foot-ground projection points, and of which height is set to the current height of CoM. Shifting to the support foot is accomplished by adding $\alpha \boldsymbol{s}_{lat}$, where $\boldsymbol{s}_{lat}$ is a unit vector for the lateral direction with respect to the pelvis frame, and $\alpha$ is the magnitude of the shift.

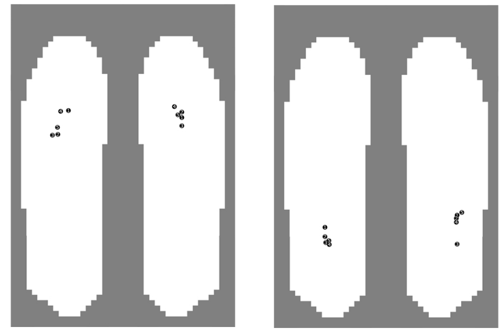



Fig. 5. Experimental result of CoP location of each foot at foot lifting induced by a large perturbation.

*Determining Swing Foot and Lifting Time:* The controller chooses the swing foot in terms of the distance between the foot and the target stepping point. In contrast to [14], we choose the foot farther to the target stepping point. This is a natural choice that prepares the robot to take additional steps if necessary. Even for single step push recovery, experiments on human in physical therapy show that 87% of predominant lateral stepping strategy is *cross over* [17], which agrees well with our choice.

Another crucial factor affecting the robot's stepping motion is the timing of foot lifting. A natural choice for this would be the moment that CoP approaches the edge of the support polygon, as used in many literatures (e.g., [12]). In fact, the validity of this approach is confirmed by out experiment on human's stepping behavior under external pushes. Figure 5 shows the location of CoPs of each foot at the instant of foot lifting when a human takes reactive stepping under forward and backward pushes. The result shows that human initiates the lift consistently when CoP reaches a certain point, which suggests that we can use the location of CoP as the indicator for foot lifting.

Based on the experiment, we set the lifting timing as the moment that CoP leaves the scaled support polygon (70% of the support polygon), and if this event occurs, the controller proceeds to the lifting phase. We also found that the *desired CoP* (Sec. IV-A) proposed by [12], instead of the actual CoP, creates more stable lifting behavior since the desired CoP is determined by the velocity-level information (i.e., momentum). Hence we replace CoP with the desired CoP ($\boldsymbol{p}_l$) for an indicator for the timing of foot lifting.

### B. Lifting Phase

When the swing foot disengages contact with the ground, the controller enters the lifting phase, in which the robot lifts the swing leg toward some target position, which is determined from the target stepping point. Researchers have developed a number of methods for determining a stepping point such as the capture point [13] and GFPE [11]. While any reasonable method should work fine, we adopted the method of [12] that determines the stepping point from the desired CoP (See Sec. IV).

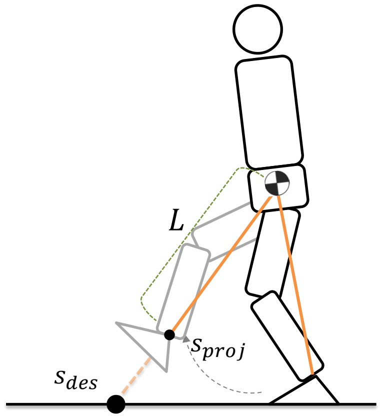



Fig. 6. An illustration of the swing phase.

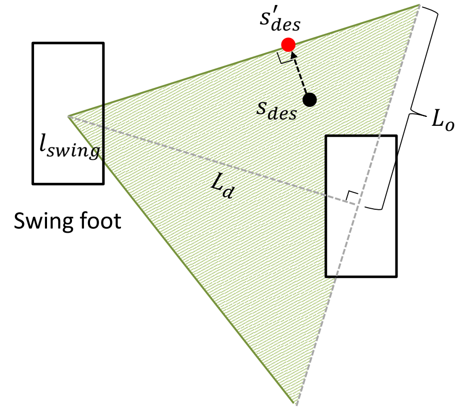



Fig. 7. Avoiding foot collision.

The target point $s_{proj}$ for the swing foot is determined as shown in Fig. 6. Given the target stepping point $s_{des}$, the vector from CoM to $s_{proj}$ is aligned to $s_{des}$ and the distance of $s_{proj}$ from CoM is set by some desired length $L$. The desired length $L$ is computed as the length from CoM to the support foot, and then down-scaled to its $80\%$ so that the robot can raise the swing foot. Using $s_{proj}$ gives an advantage that the controller does not need to consider the trajectory of the swing foot while generating reasonably natural stepping motion. Hence, the desired swing foot configuration $\boldsymbol{T}_d \in SE(3)$ is determined such that the position equals $s_{proj}$ and the orientation is set to identity. The desired swing foot velocity $\boldsymbol{v}_d$ is simply set to zero. The controller updates $s_{des}$ repeatedly at every control time step during the lifting and landing phases to adapt to continuous perturbations.

During the lifting and the subsequent landing phases, linear momentum remains passive by specifying inputs as follows:

$$\boldsymbol{l}_d = \boldsymbol{l} \tag{5}$$

$$\boldsymbol{r}_{Gd} = \boldsymbol{r}^*_{Gd} \tag{6}$$

Note that $\boldsymbol{r}^*_{Gd}$ is updated at every control time step and thus it translates adaptively to the pushed direction. Therefore, rather than “holding back” CoM toward the support foot, $\boldsymbol{r}_{Gd}$ allows for the smooth transition of CoM.

When the swing foot gets close enough to $s_{proj}$ (within 2 centimeters in our experiment), the controller proceeds to the landing phase.

### C. Landing Phase

The landing phase is similar to the lifting phase. The only difference is that now the length parameter $L$ is set to the distance from CoM to the support foot (not down-scaled). When the swing foot touches the ground, the stepping controller ends its task and a higher level controller determines whether to resume the reactive stepping controller or to trigger the postural balance controller. Note that there can be discrepancy between the actual stepping location and $s_{des}$. However, the discrepancy is not significant for regular stepping.

### D. Avoiding Foot Collision

During the lifting and landing phases, it is important to avoid the collision of the swing foot to the environment or other body parts. While it is a challenging problem to develop a generalized algorithm to avoid all possible collisions, at least avoiding collisions between the two feet helps the controller cope with perturbations in larger range of directions. In particular, when a lateral perturbation is applied to a robot, the swing foot should turn around the support foot to prevent collision. To enable such a behavior, we employ a simple rule; that is, the controller prevents the swing foot from being too close to the support foot by adjusting $s_{des}$.

To this end, we define an available region that the swing foot can be placed. Figure 7 illustrates the process. Starting from the swing foot-ground projection position $l_{swing}$, we draw a line to the center of the support foot. Then given some safety distance $L_0$ from the support foot, an isosceles triangle is defined such that the height is $L_d$ and the length of the base is $2L_0$. If $s_{des}$ is included in the triangle, it is projected to the nearest point $s'_{des}$ outside the triangle. This simple rule is effective when the perturbation is applied from the diagonal direction; without this evasive maneuver, the swing foot easily collides with the support leg.

## IV. Simulation Experiment

We tested the reactive stepping controller with a simulated humanoid robot model, which consists of 16 joints with 39 DoFs. It is approximately 56 Kg in weight and 1.5 meters in height (Fig. 8). Simulation is performed using Bullet physics engine (www.bulletphysics.com).

### A. Indicators for Control Trigger, Lifting, and Target Stepping Point

We use the *desired CoP* indicator for deciding foot lifting timing, determining the target step point as well as triggering the stepping controller, as proposed by [12]. The desired CoP is the imaginary CoP given the desired rate of change of linear and angular momenta.

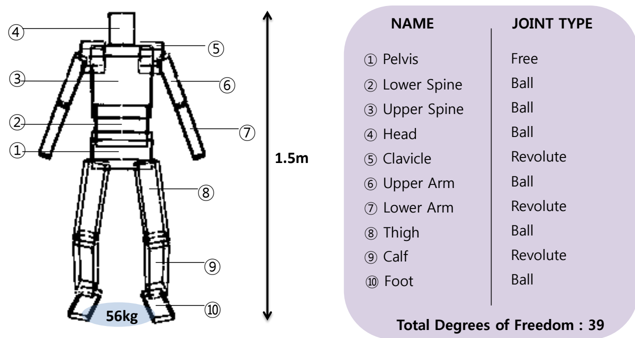



Fig. 8. Humanoid robot model specification.

Let us first recall the CoP $(p_x, p_y)$ equations:

$$p_x = r_{G,x} - \frac{\dot{l}_x}{f_y} r_{G,y} + \frac{\dot{k}_z}{f_y} \tag{7}$$

$$p_z = r_{G,z} - \frac{\dot{l}_z}{f_y} r_{G,y} - \frac{\dot{k}_x}{f_y}, \tag{8}$$

where $f_y = \dot{l}_y - mg$ is the vertical ground reaction force. If we set $\boldsymbol{l}$ and $\boldsymbol{k}$ to the desired linear and angular momentum rate change, the resulting CoP represents the desired CoP corresponding to the desired momenta rate change. Following [12], we set the desired momenta rate changes such that they stabilize the current linear and angular momenta: $\dot{\boldsymbol{l}} = -d_l \boldsymbol{l}$ and $\dot{\boldsymbol{k}} = -d_k \boldsymbol{k}$. Then the desired CoP $\boldsymbol{p}_d$ is computed as follows:

$$p_{d,x} = r_{G,x} + \frac{d_{l,h} l_x}{f_y} r_{G,y} - \frac{d_{k,h} k_z}{f_y} \tag{9}$$

$$p_{d,z} = r_{G,z} + \frac{d_{l,h} l_z}{f_y} r_{G,y} + \frac{d_{k,h} k_x}{f_y} \tag{10}$$

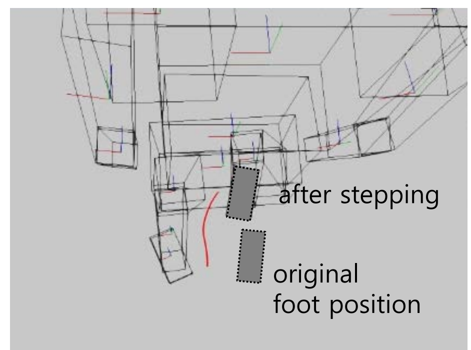
Since $\boldsymbol{p}_d$ are determined from the momentum information, its location changes somewhat smoother than the actual CoP. Note that the location of $\boldsymbol{p}_d$ is determined by the gain parameters $d_l, d_k$. If they are set high, $\boldsymbol{p}_d$ is located farther from the foot.

By setting different gains to (9) and (10), we use the desired CoP's for determining the triggering of stepping controller, lifting timing, and target stepping point. If the desired CoP for control trigger leaves the support polygon, the high level controller invokes the reactive stepping controller. Subsequently if the desired CoP for lifting timing leaves the down-scaled support polygon, the controller enters the lifting phase. Obviously, the desired CoP for target stepping point should be located far enough from the support polygon. Therefore, it is reasonable that the gains for foot lifting should be less than those for the control trigger, which should be less than those for the target stepping point. Specifically, from simulation trials, we set $d_l = 4$ and $d_k = 4$ for control trigger, $d_l = 2$ and $d_k = 2$ for foot lifting timing, and $d_l = 4$ and $d_k = 25$ for the target stepping point. Note that our stepping controller is not restricted to using the desired CoP indicators. Other indicators such as capture point or GFPE can be used as well.

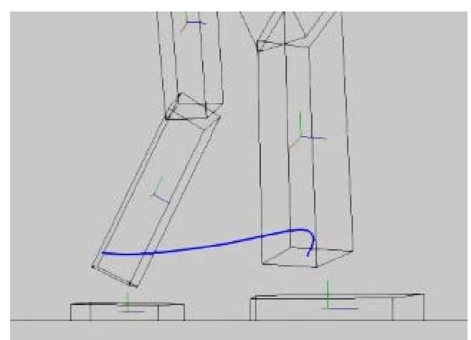
### B. Push Recovery

In simulation tests, we investigated the performance of the reactive stepping controller by applying perturbations in varying directions and magnitudes to the humanoid robot model (Figs. 9-11). Fig. 9 shows the different step lengths per different push strengths. When perturbed by a higher force, the robot naturally takes a longer step.

Our controller can deal with perturbations in diagonal directions (Fig. 10). Fig. 10 (a) shows that our controller can take multiple steps if a single step cannot fully suppress the instability. The endurable range of magnitude of diagonal push is more limited than that of forward push because the discrepancy between the target stepping point and the actual stepping location increases due to the collision-avoiding maneuver of the swing foot explained in Sec. III-D. However, without the collision avoidance strategy, the swing foot easily bumps against the support leg, leading to the toppling of the robot as shown in Fig. 10 (b).



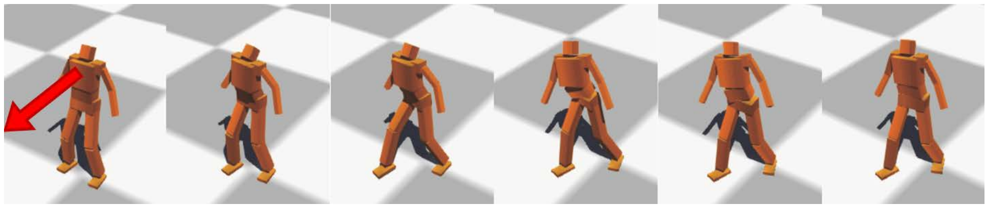

(a) CoM trajectory (red curve) seen from the top



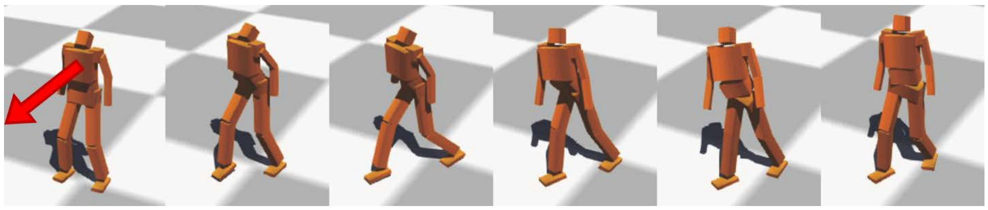
(b) The trajectory of swing foot (right) in sagittal plane

Fig. 12. CoM and swing foot trajectories during forward stepping.

Fig. 12 shows the trajectories of CoM and the swing foot created by the stepping controller. It demonstrates that smooth curves emerge from our controller. The CoM trajectory is curved to the support foot to prevent toppling sideways, and it is similar to the trajectory of the well-known inverted pendulum model. The swing foot trajectory shows a smooth curve, but the overall shape is somewhat different from the hill-shaped curves used in other literatures [11], [12]. This implies that, although the current stepping controller is working quite effectively without explicit specifica-

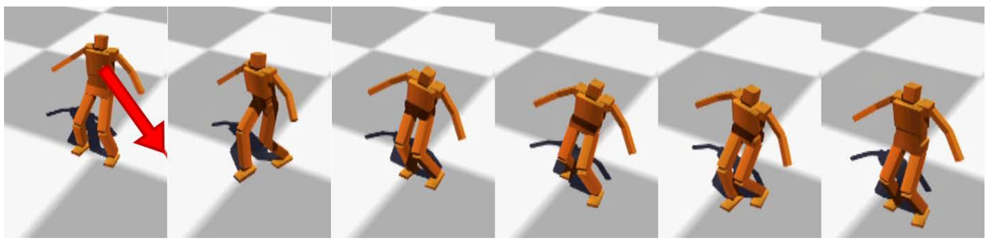


(a) Forward push, 120N, 0.5s

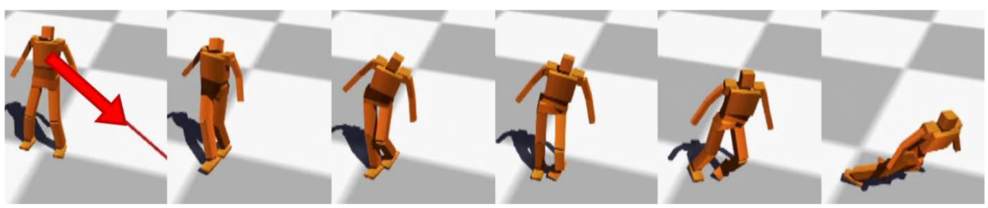


(b) Forward push, 170N, 0.5s

Fig. 9. Reactive stepping motion against forward pushes.



(a) With avoidance of foot collision



(b) Without foot collision avoidance

Fig. 10. Stepping motions against perturbations (120N, 0.5s) in diagonal direction.

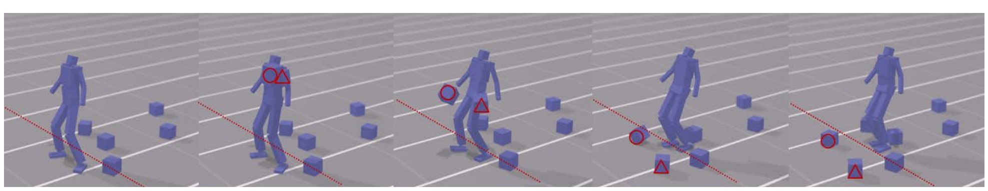


Fig. 11. Two boxes (marked with a circle and a triangle) hit the robot's trunk and the robot takes multiple steps backward to maintain balance.

tion of the trajectories, the resulting stepping motion may be different to a certain extent from those generated by previous methods.

The robustness of the controller is demonstrated by numerous impacts from the boxes thrown from arbitrary directions (Fig. 11). The robot uses both postural balance control and reactive stepping strategy to retain the balance. The robot adaptively recover its balance from arbitrary perturbations, creating even backward stepping behaviors when pushed backward by the boxes.

## V. Conclusion

In this paper, we introduced a novel reactive stepping controller using momentum control. Instead of specifying the desired trajectories of CoM and the swing foot, our controller generates reactive stepping motion passively from the momentum induced by the external perturbations. The major advantage of this approach is that the controller need not attempt to follow some prescribed trajectory, which is typically not optimal from the perspective of dynamics, and gains more adaptive behavior by yielding to the external perturbations. Constructed on top of the momentum controller, our reactive stepping controller is realized very straightforwardly by setting suitable inputs for the momentum controller in each phase.

There are many venues for future work. In this work, some parameter values such as CoM shift ($\alpha$ in (4)) and the desired length of the swing leg during the lifting phase are determined heuristically from experiments. A technique to set those parameters adaptively based on the direction and magnitude of a perturbation will increase the performance of the stepping controller. Currently, the upper body only maintains the default pose. However, the reactive stepping controller may become more efficient if it involves the upper body movements (e.g., arm swinging). Also, a robust stepping controller should be applicable not only to the flat ground, but also to more complex environments such as non-level ground and the ground with many obstacles.

## REFERENCES


[1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momen-tum control: Humanoid motion planning based on the linear and angular momentum," *IEEE/RSJ Intn'l Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[2] M. Abdallah and A. Goswami, "A biomechanically motivated two-phase strategy for biped upright balance control," in *IEEE Intn'l Conf. on Robotics and Automation (ICRA)*, 2005, pp. 2008–2013.

[3] Y. Abe, M. Da Silva, and J. Popović, "Multiobjective control with frictional contacts," *ACM SIGGRAPH/EG Symposium on Computer Animation*, pp. 249–258, 2007.

[4] A. Macchietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 80:1–80:8, Jul. 2009.

[5] T. Sugihara, "Mobility enhancement control of humanoid robot based on reaction force manipulation via whole body motion," *PhD Thesis, University of Tokyo*, Oct. 2004.

[6] T. Komura, A. Nagano, H. Leung, and Y. Shinagawa, "Simulating pathological gait using the enhanced linear inverted pendulum model," *IEEE Transactions on Biomedical Engineering*, vol. 52, pp. 1502–1513, Sep. 2005.

[7] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, vol. 33, no. 4, pp. 399–414, Apr. 2012.

[8] A. Sano and J. Furusho, "Realization of natural dynamic walking using the angular momentum information," *IEEE Intn'l Conf. on Robotics and Automation (ICRA)*, 1990.

[9] B. Stephens, "Integral control of humanoid balance," in *IEEE/RSJ Intn'l Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[10] B. Maki, W. McIlroy, and G. Fernie, "Change-in-support reactions for balance recovery," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 22, no. 2, pp. 20 – 26, march-april 2003.

[11] S.-K. Yun and A. Goswami, "Momentum-based reactive stepping controller on level and non-level ground for humanoid robot push recovery," in *IEEE/RSJ Intn'l Conf. on Intelligent Robots and Systems (IROS)*, Sept. 2011, pp. 3943 –3950.

[12] C.-C. Wu and V. Zordan, "Goal-directed stepping with momentum control," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, 2010, pp. 113–118.

[13] J. Pratt, S. Carff, Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," *Humanoids2006*, Dec. 2006.

[14] J. Wu and Z. Popović, "Terrain-adaptive bipedal locomotion control," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 72:1–72:10, Jul. 2010.

[15] T. McGeer, "Passive dynamic walking," *International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[16] S.-H. Hyon and T. Emura, "Symmetric walking control: Invariance and global stability," in *IEEE Intn'l Conf. on Robotics and Automation (ICRA)*, April 2005, pp. 1455 – 1462.

[17] B. E. Maki and W. E. McIlroy, "The role of limb movements in maintaining upright stance: the "change-in-support" strategy." *Phys Ther*, vol. 77, no. 5, pp. 488–507, 1997.