2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Unified Balance Control for Biped Robots Including Modification of Footsteps with Angular Momentum and Falling Detection Based on Capturability

Yuta Kojio, Yasuhiro Ishiguro, Kim-Ngoc-Khanh Nguyen,
Fumihito Sugai, Yohei Kakiuchi, Kei Okada and Masayuki Inaba

Abstract— **In this paper, we propose walking balance control based on Caputurability. The proposed method consists of five strategies: (i) moving Zero Moment Point (ZMP) in the support polygon (ii) landing position modification (iii) landing timing modification (iv) angular momentum control (v) falling detection and fall control. Walking pattern generation calculates the ZMP so that the Capture Point (CP) reaches the position of the supporting foot at the end of the double support phase. Owing to the asymmetry of the reachable landing region, landing timing modification is different in the sagittal and lateral planes, and the step time is extended in the lateral plane depending on the direction of disturbances. The torque around the center of gravity to avoid falling is realized through whole-body inverse kinematics with constraints on the angular momentum. In addition, we propose falling detection considering the reachable landing region. We verified the effectiveness of the proposed method through experiments in which the biped robot was disturbed by pushing during tether-free walking. The robot could prevent breakdown by detecting possible falling and performed knee bending motions to suppress damage.**

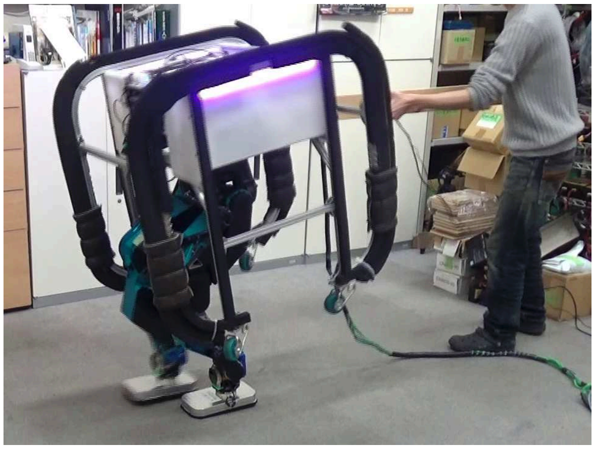


Fig. 1. Robot that is pushed forward

I. INTRODUCTION

Humanoid robots have the potential to move in a variety of environments because they have a form similar to humans. However, as it is difficult to completely simulate such environments, it is inevitable that the robot is disturbed while walking (Fig.1). Therefore, balancing control to prevent falling even under disturbances has been studied extensively [1]–[4]. Most previous studies stabilized walking by moving the Zero Moment Point (ZMP). Some studies have also proposed methods in which the robot changes the support polygon; that is to say, it modifies the landing position [5], [6]. In addition, the robot can maintain balance by modifying the landing timing as well as position [7], [8]. The robot can deal with larger disturbances by using redundant degrees of freedom and controlling angular momentum around the center of gravity (COG) while stepping [9]–[11]. Modifying the landing timing or controlling the angular momentum not only increases acceptable magnitude of external forces but also enables the robot to deal with disturbances even in narrow footholds instead of long distance stepping.

Pratt et al. [12] presented the Capture Point (CP), an asymptotic point at which the COG of linear inverted pendulum model (LIPM) [13] converges. The robot does not fall if the CP is inside the support polygon, and thus, the CP is compatible with the landing position. Koolen et al. [14] proposed N-step Capturability that indicates how many steps are necessary for the COG to converge to the stable state.

Sugihara et al. [15] proposed a walking pattern generation method whose constraint includes only the "standing stability" when landing. It can flexibly deal with disturbances because the landing position and timing of the next step are the only walking commands, and the position and velocity of the COG are required for observation. Additionally, its computational cost is low, and Yamamoto et al. [16] extended the method so that the robot modifies the landing position and timing depending on disturbances. However, it assumes that the reachable landing region is circular and centered at the supporting foot although the region is not symmetrical owing to leg interference. Therefore, for instance, it cannot properly deal with external forces to the right when the right foot is the supporting foot. Moreover, it approximates the sole as a circle, which is not suitable for the robot whose foot sole is a vertically long rectangle.

Although there are many studies on balance control, falling is inevitable. Therefore, fall control to absorb the impact during falling, such as touching an arm to the ground, has been also studied [17]–[19]. To create enough time to change to the fall pose, it is desirable to detect falling as soon as possible. However, detecting falling too early deteriorates the walking ability of the robot, and therefore, detection must be

Y. Kojio, Y. Ishiguro, K. Nguyen, F. Sugai, Y. Kakiuchi, K. Okada and M. Inaba are with Department of Mechano-Infomatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan kojio at jsk.imi.i.u-tokyo.ac.jp

978-1-7281-4003-2/19/$31.00 ©2019 IEEE

suitable for balance control.

II. Walking pattern generation based on foot-guided agile control

A. Foot-guided agile control

We generate a walking pattern based on foot-guided agile control proposed by Sugihara et al. [15]. In this section, we briefly describe the method.

The motion of the COG of the LIPM in the sagittal plane is obtained as follows:

$$\ddot{x}_G = \omega^2(x_G - x_Z) \tag{1}$$

$$\omega := \sqrt{\frac{g}{z_G - z_Z}} \tag{2}$$

where x, y, z are the components of the position vector and subscripts G, Z denote the COG and ZMP, respectively. g is the gravitational acceleration.

If the landing position corresponds to the CP, standing stability is guaranteed when landing. The optimization problem for keeping the ZMP as close to the supporting foot as possible is expressed as follows:

$$x_Z(t) = \arg\min_{x_Z} \frac{1}{2}\int_t^T (x_Z(\tau) - x_Z^{ref})^2 d\tau \tag{3}$$

$$s.t. \quad x_{cp}(T) = x_f \tag{4}$$

where

$$x_{cp}(t) = x_G(t) + \frac{\dot{x}_G(t)}{\omega} \tag{5}$$

$$x_Z^{ref} = x_{sup} \tag{6}$$

T is the time to land, x_f is the landing position, and x_{sup} is the position of the supporting foot. The ZMP satisfying the problem above is obtained as follows:

$$x_Z(t) = x_{sup} + \frac{2({}^{sup}x_{cp}(t) - {}^{sup}x_f e^{-\omega(T-t)})}{1 - e^{-2\omega(T-t)}} \tag{7}$$

$${}^{sup}x_{cp}(t) := x_{cp}(t) - x_{sup} \tag{8}$$

$${}^{sup}x_f := x_f - x_{sup} \tag{9}$$

B. Extension for double support phase

The method in the preceding section assumes that the supporting foot is exchanged instantaneously. To reduce the landing impact and smoothen the walking motion during application to real robots, we extend the method to take the double support phase into account.

Considering T_s as the time when the current swinging foot lands and T_d as the time when the current supporting foot lifts off the ground, we define the reference ZMP $x_Z^{ref}(t)$ in the double support phase as the following linear function:

$$x_Z^{ref}(t) = at + b \tag{10}$$

$$a := \frac{x_Z^{next} - x_Z^{cur}}{T_d - T_s} \tag{11}$$

$$b := x_Z^{cur} - \frac{x_Z^{next} - x_Z^{cur}}{T_d - T_s} T_s \tag{12}$$

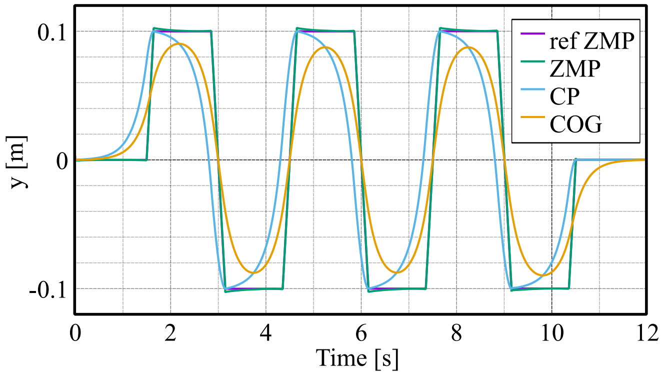


Fig. 2. Generated trajectory. The step length is 0.2 m and walking cycle is 1.5 s, 20% of which is the double support phase. The output ZMP is generated so that the CP reaches the supporting foot at the end of the double support phase.

where x_Z^{cur}, x_Z^{next} are the constant reference ZMP of the current and next step respectively. We generate the ZMP so that the next supporting foot position corresponds to the CP when the current supporting foot lifts off. The optimization problem for keeping the ZMP as close to the reference ZMP as possible is expressed as follows:

$$x_Z(t) = \arg\min_{x_Z} \frac{1}{2}\int_t^{T_d} (x_Z(\tau) - x_Z^{ref}(\tau))^2 d\tau \tag{13}$$

$$s.t. \quad x_{cp}(T_d) = x_f \tag{14}$$

where

$$x_{cp}(t) = x_G(t) + \frac{\dot{x}_G(t)}{\omega} \tag{15}$$

$$x_Z^{ref}(t) = \begin{cases} x_Z^{cur} & (t \le T_s) \\ at + b & (otherwise) \end{cases} \tag{16}$$

The problem stated above is solved as follows:

$$x_Z(t) = \begin{cases} x_Z^{ref}(t) + \dfrac{2({}^{Z}x_{cp}(t) - {}^{Z}x_f e^{-\omega(T_d-t)})}{1 - e^{-2\omega(T_d-t)}} \\ \quad + \dfrac{\frac{2a}{\omega}(e^{-\omega(T_d-t)} - e^{-\omega(T_s-t)})}{1 - e^{-2\omega(T_d-t)}} & (t \le T_s) \\ x_Z^{ref}(t) + \dfrac{2({}^{Z}x_{cp}(t) - {}^{Z}x_f e^{-\omega(T_d-t)})}{1 - e^{-2\omega(T_d-t)}} \\ \quad + \dfrac{\frac{2a}{\omega}(e^{-\omega(T_d-t)} - 1)}{1 - e^{-2\omega(T_d-t)}} & (otherwise) \end{cases} \tag{17}$$

$${}^{Z}x_{cp}(t) := x_{cp}(t) - x_Z^{ref}(t) \tag{18}$$

$${}^{Z}x_f := x_f - x_Z^{ref}(T_d) = x_f - x_Z^{next} \tag{19}$$

As Eq.(13) does not include constraints on the support polygon, the ZMP obtained by Eq.(17) can be outside the region. If the ZMP is outside the region, we project it to the nearest point on the edge of the region. The COG trajectory is derived from the ZMP as follows:

$$\frac{d}{dt}\begin{pmatrix} x_G \\ \dot{x}_G \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \omega^2 & 0 \end{pmatrix}\begin{pmatrix} x_G \\ \dot{x}_G \end{pmatrix} + \begin{pmatrix} 0 \\ -\omega^2 \end{pmatrix} x_Z \tag{20}$$

Fig.2 shows the y component of the generated trajectory for walking forward. The step length is 0.2 m and walking

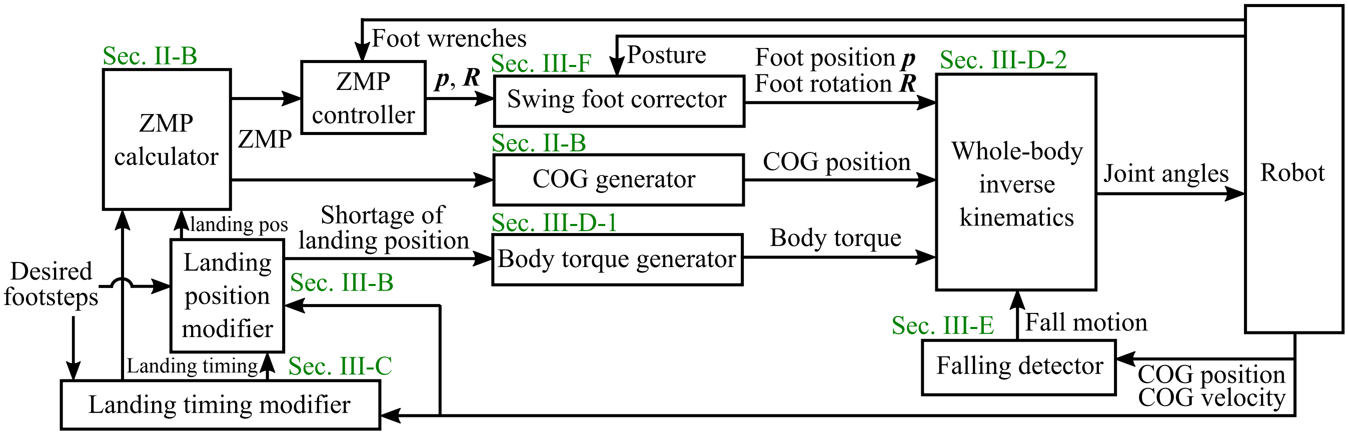


Fig. 3. System architecture of the proposed controller

cycle is 1.5 s, 20% of which is the double support phase. As seen in Fig.2, the generated ZMP almost followed the reference ZMP and CP reached the supporting foot position at the end of the double support phase.

III. Adaptation to Large Disturbances

Approximating the foot shape to be rectangular, the realized ZMP is constrained within the support polygon as follows:

$$x_Z^{min} \le x_Z \le x_Z^{max} \tag{21}$$

Therefore, there is a limit for handling disturbances using the method described in II. In this section, we describe the modification of the landing position and timing, angular momentum control, and falling detection methods to deal with larger disturbances. In addition, we introduce a swinging foot correction method to realize the desired landing position under disturbances.

A. System overview

Fig.3 shows an overview of the proposed controller. The robot control the ZMP, landing position, landing timing, and angular momentum in this order as disturbances become larger. When the robot cannot maintain balance using these strategy, the robot stops walking and performs the fall motion to prevent damage.

B. Landing position modification

The differential of the CP is obtained from Eq.(1) and Eq.(5).

$$\dot{x}_{cp} = \omega(x_{cp} - x_Z) \tag{22}$$

Assuming that the ZMP reaches the edge of the support polygon and is constant, CP at T_d is solved from Eq.(22).

$$x_{cp}(T_d) = e^{\omega(T_d - t)}(x_{cp}(t) - x_Z) + x_Z \tag{23}$$

The robot's walking is stabilized by moving the ZMP within the support polygon based on the following condition obtained from Eq.(21) and Eq.(23).

$$x_Z^{min} \le \frac{x_f - e^{\omega(T_d - t)} x_{cp}(t)}{1 - e^{\omega(T_d - t)}} \le x_Z^{max} \tag{24}$$

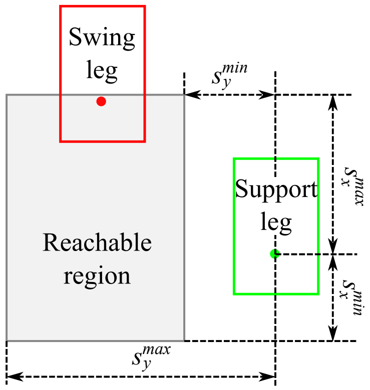


Fig. 4. Reachable landing region. The swinging foot can land in the shaded region. If the modified landing position is outside the region, it is projected to the region. s is the parameter for stride limitation, which is relative to the supporting foot.

If this condition is not satisfied, walking can be stabilized at T_d by modifying the reference landing position. In the case where the robot is disturbed forward, the modified landing position $\acute{x}_f$ is obtained from Eq.(24) as follows:

$$\acute{x}_f = e^{\omega(T_d - t)}(x_{cp} - x_Z^{max}) + x_Z^{max} \tag{25}$$

We limit the reachable region in which the swinging foot can land as shown in Fig.4.

C. Landing timing modification

Owing to the limitations regarding the modified landing position, there is also a limit for handleable disturbances even with a landing position modification. At such times, the robot can be stabilized by extending or shortening the time remaining before landing.

We calculate the temporary landing position from Eq.(25) using the desired landing timing. If the position is outside the reachable region, the landing timing needs to be changed. $\acute{x}_f$ is moved to the edge of the region or gets close to it by modifying the step time.

Because of leg interference, the robot generally cannot step in the lateral direction over the supporting foot. If the

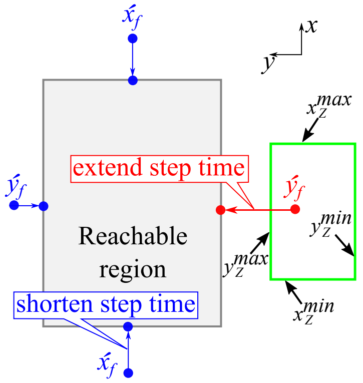


Fig. 5. Step time modification. If the position is outside the reachable region, the landing timing needs to be changed. If the temporary modified landing position is on the right side of the reachable region, the step time is extended. Otherwise, it can be shortened.

temporary landing position $\acute{y}_f$ is on the right side of s_y^{min}, the step time is extended for stabilizing the robot. Otherwise, the step time can be shortened as shown in Fig.5.

1) Shortening step time: If the robot is disturbed forward, it is necessary to shorten the step time based on the following condition:

$$s_x^{max} < \acute{x}_f \tag{26}$$

The modified landing time $\acute{T}_d$ is derived from Eq.(23).

$$\acute{T}_d = t + \frac{1}{\omega}\ln\frac{s_x^{max} - x_Z^{max}}{x_{cp}(t) - x_Z^{max}} \tag{27}$$

2) Extending step time: If the robot is disturbed to the right while the right foot is supporting it, the step time is extended based on the following condition:

$$\acute{y}_f < s_y^{min} \wedge y_Z^{min} < y_{cp} \tag{28}$$

The modified landing time is obtained as follows:

$$\acute{T}_d = t + \frac{1}{\omega}\ln\frac{s_y^{min} - y_Z^{min}}{y_{cp}(t) - y_Z^{min}} \tag{29}$$

Note that the robot's walking cannot be stabilized in one step when $y_Z^{min} \geq y_{cp}$. At such times, it is required to land as soon as possible.

The lower value of the desired modifying time on the x axis and y axis is used as the modified time $\acute{T}_d$. The modified step time is limited to T^{min}, and the modified remaining time before landing is limited to T^{mgn}, at least 0. Thus, $\acute{x}_f$ calculated using the modified time $\acute{T}_d$ can be outside the reachable region. Moreover, landing time modification influences the state in the other axis. Therefore, we modify the landing timing first and then the landing position if necessary. If the modified landing position is outside the reachable region, it is projected to the region. The shortage of the position is compensated by using the angular momentum control described in the next section.

D. Angular momentum control

When the robot rotates around the COG, Eq.(23) is expressed using the Centroidal Moment Pivot (CMP) [20] as follows:

$$x_{cp}(T_d) = e^{\omega(T_d - t)}(x_{cp}(t) - x_{cmp}) + x_{cmp} \tag{30}$$

Considering the torque around the COG to be $\boldsymbol{\tau} = (\tau_x\ \tau_y\ \tau_z)^T$, the relationship between the ZMP and CMP is given as follows:

$$x_{cmp} = x_Z + \frac{\tau_y}{Mg} \tag{31}$$

Therefore, generating the torque compensates for the shortage of the ZMP owing to the limitations regarding the step time and landing position.

1) Torque around COG generation: From Eq.(30), the relationship between the modified landing position and CMP is obtained as follows:

$$x_{cmp} = \frac{\acute{x}_f - e^{\omega(\acute{T}_d - t)}x_{cp}(t)}{1 - e^{\omega(\acute{T}_d - t)}} \tag{32}$$

Therefore, the CMP that compensates for the shortage Δx_f of the landing position is given as follows:

$$\Delta x_{cmp} = \frac{\Delta x_f}{1 - e^{\omega(\acute{T}_d - t)}} \tag{33}$$

Then, torque compensation τ_y is derived as follows:

$$\tau_y = Mg\Delta x_{cmp} = \frac{Mg\Delta x_f}{1 - e^{\omega(\acute{T}_d - t)}} \tag{34}$$

Torque compensation is calculated in both the single and double support phases although the landing position and timing are modified only in the single support phase. Hence, instead of using Eq.(33), Δx_{cmp} in the double support phase is replaced by Δx_{cmp} just before the double support phase.

2) Whole-body inverse kinematics considering angular momentum: We solve inverse kinematics (IK) using the Levenberg–Marquardt method proposed in [21]. In this paper, the constraints are the (i) foot position, (ii) foot orientation, (iii) root link orientation, (iv) COG position, and (v) angular momentum. The reference angular momentum $\boldsymbol{L}_G$ for IK generating rotation around the COG to maintain balance is given as follows:

$$\boldsymbol{L}_G = \boldsymbol{L}_G^{gen} + \boldsymbol{\tau}\Delta t \tag{35}$$

where $\boldsymbol{L}_G^{gen}$ is the angular momentum that is generated from the previous IK. We limit the magnitude of the root link posture $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^{lim}$.

Because of the limit imposed on the root link posture, the robot cannot always generate the reference angular momentum. Therefore, the robot is desired to return to the reference posture when the disturbances are small. Thus, we set the weight for the angular momentum to a large value if the torque $\boldsymbol{\tau}$ for balancing is needed; otherwise, we set it to a small value. We interpolate the weight using the minimum jerk polynomial [22] so that the joint angle does not drastically change.

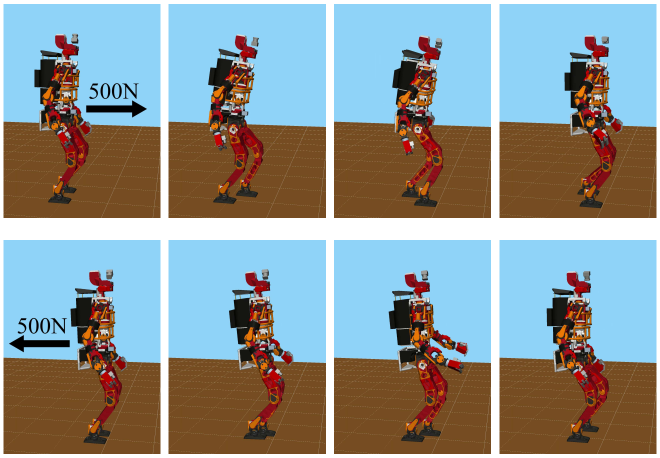


Fig. 6. Simulation result in which the robot with arms was pushed forward (top) and backward (bottom) with 500N/0.1s. The robot generated the angular momentum for recovering balance, which was realized using whole-body inverse kinematics.

Fig.6 shows the simulation result obtained using Choreonoid [23] in which the humanoid robot with arms JAXON [24] is pushed forward and backward. An external force of 500 N was applied to the upper body for 0.1 s. The robot withstood the disturbances by mainly swinging its arms for generating angular momentum.

E. Falling detection

Even if the angular momentum and landing position/timing are modified, extremely large disturbance causes the robot to fall. For larger disturbances, fall control is effective in suppressing damage to the robot. Therefore, it is better to detect falling as quickly as possible.

Assuming that the angular momentum cannot stabilize walking any more if the posture angle is close to the limit θ^{lim}, we determine whether it is possible to return to stable states via multiple stepping.

1) Sagittal plane: In the sagittal plane, if the CP relative to the supporting foot when the current supporting foot lifts off is greater than the CP relative to the foot when the next foot lifts off, the robot can stabilize walking via multiple stepping. If the robot is disturbed forward, the condition is expressed as follows:

$$x_{cp}(T_d) - x_f > x_{cp}(T_d + T^{min}) - (x_f + s_x^{max}) \quad (36)$$

$$\Leftrightarrow \quad x_{cp}(T_d) < x_Z^{max} - \frac{s_x^{max}}{1 - e^{\omega T^{min}}} \quad (37)$$

$x_{cp}(T_d)$ is calculated from Eq.(23). If the condition Eq.(37) is not satisfied, the robot will fall down.

2) Lateral plane: In the lateral plane, the reachable landing position is asymmetrical because of leg interference. Therefore, the condition for converging requires the CP in two steps. If the robot is disturbed to the right while the right foot is supporting, the condition is expressed as follows:

$$y_{cp}(T_d) - y_f > y_{cp}(T_d + 2T^{min}) - (y_f + s_y^{max} - s_y^{min}) \quad (38)$$

$$\Leftrightarrow \quad y_{cp}(T_d) < \frac{y^{in} + y^{out} e^{\omega T^{min}}}{1 + e^{\omega T^{min}}} - \frac{s_y^{max} - s_y^{min} e^{\omega T^{min}}}{1 - e^{2\omega T^{min}}} \quad (39)$$

$$y^{in} := |y_Z^{max} - y_{sup}| \quad (40)$$

$$y^{out} := |y_Z^{min} - y_{sup}| \quad (41)$$

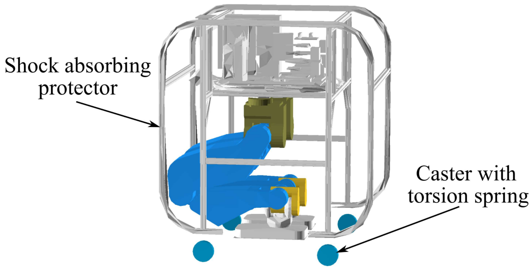


Fig. 7. Knee bending motion for fall control. The robot's link does not contact the ground even if the robot rotates on the ground because the protector covers all the links in this pose.

F. swinging foot correction

If the robot tilts owing to disturbances, the swinging foot may deviates from desired trajectory. This results in an excessive landing impact or an error in the reference landing position. The former may be absorbed by fast force control. On the other hand, the latter cannot be resolved using the force control, thus affecting the balance of the robot. Therefore, considering an actual robot's posture, we correct the swinging foot position and rotation as follows:

$$\Delta\dot{\boldsymbol{p}} = k(\boldsymbol{p}_{act} - \boldsymbol{p}_{ref}) \quad (42)$$

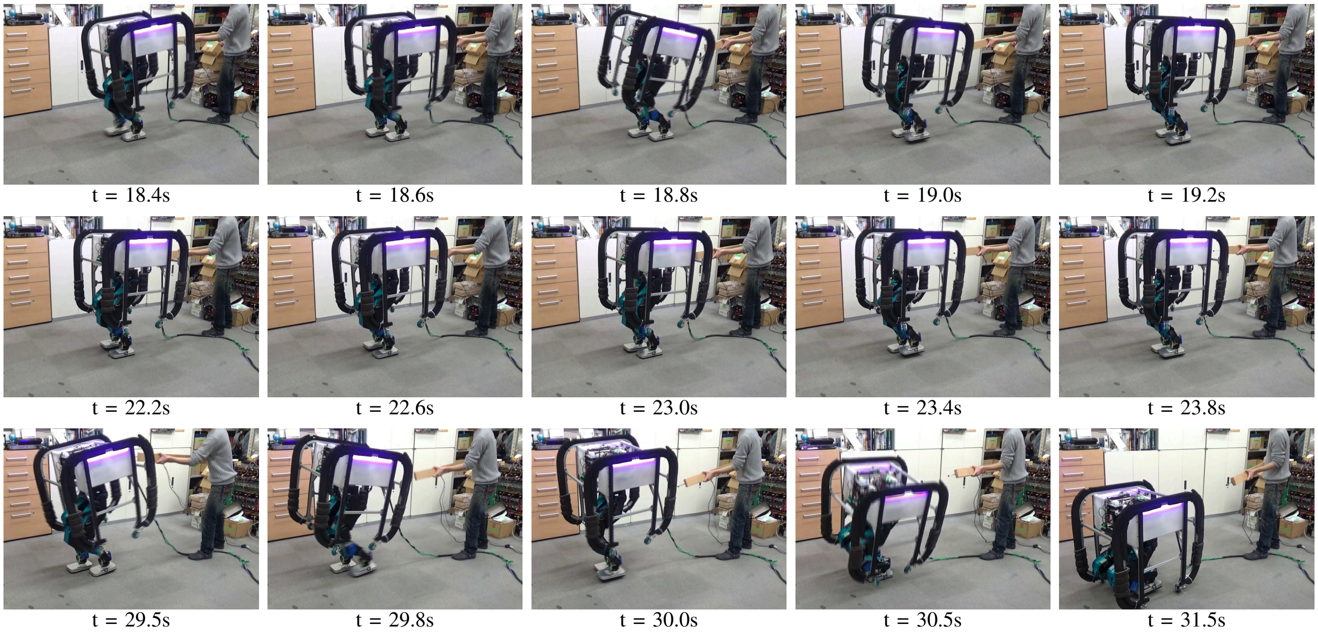
$$\Delta\boldsymbol{\Omega} = k(\boldsymbol{R}_{ref}^T \boldsymbol{R}_{act})^{\vee} \quad (43)$$

where $\boldsymbol{p}$, $\boldsymbol{R}$ are the swinging foot position and rotation relative to the supporting foot. $\boldsymbol{\Omega}$ is the angular velocity vector and $\vee$ represents the operator that derives the vector from a skew-symmetric matrix. Subscripts ref, act respectively denote the reference and actual values calculated using a robot's posture and joint angles. k is the gain and was set to 10 in this study. Although this paper is not centered on this control, it was effective in order to realize the desired landing position and suppress a landing impact.

IV. Experimental Results

A. Hardware and fall control

We validated the proposed methods through the experiments using a biped robot CHIDORI-F. The robot is 1.2 m tall, weighs 67 kg and has a shock absorbing protector around the body. In addition, it has a six-axis force sensor in each ankle, and a gyro and acceleration sensor in its body. The



t = 18.4s t = 18.6s t = 18.8s t = 19.0s t = 19.2s

t = 22.2s t = 22.6s t = 23.0s t = 23.4s t = 23.8s

t = 29.5s t = 29.8s t = 30.0s t = 30.5s t = 31.5s

Fig. 8. Experiment for forward push recovery while walking on the spot. Top: peak 68N; impulse 26.5Ns; the robot modified the landing position/timing and angular momentum. Middle: peak 105N; impulse 20.4Ns; the robot modified the landing position. Bottom: peak 116N; impulse 36.0Ns; the robot detected possible falling to the left and bent the knees.

robot can prevent damage by bending its knees as shown in Fig.7 because every link does not contact the ground when falling, which enables us to conduct experiments without a tether including the start-up of the robot. In addition, casters with torsion springs are equipped at the bottom of the protector, which help absorb shock when falling and move on the ground.

The robot stops walking and bends its knees with a minimum jerk polynomial interpolation over 1 s, soon after detecting falling using the method described in III-E.

B. Forward push recovery

Fig.8 shows the snapshots of the forward push recovery experiment. The robot was pushed forward by a bar with a force sensor while walking on the spot at a nominal step cycle of 1.2 s. Fig.9(a) shows the external force, Fig.9(b) and Fig.9(c) show the generated ZMP, actual CP, and modified landing position; Fig.9(d) shows the modified landing timing, and Fig.9(e) shows the torque around the COG. Fig.10 shows a zoomed view of τ_y at approximately 24.9 s in Fig.9(e).

The robot was stabilized by moving the ZMP within the support polygon when the influence of the external force was small at approximately 20.5 s. The peak force was 68 N and the impulse was 17.7 Ns ((i) in Fig.9).

In the middle part of Fig.8, the peak force was 105 N and the impulse was 20.4 Ns ((ii) in Fig.9). The robot modified the landing position because the force was larger. Fig.9 shows that the actual CP approximately reached the position of the supporting foot at the time of foot exchange.

In the top part of Fig.8, the peak force was 68 N and the impulse was 26.5 Ns ((iii) in Fig.9). The robot first modified the landing position/timing and angular momentum for the x component, and then modified the landing position/timing for the y component at the second step. It is considered that the reason why the robot took a step to the left despite being pushed forward is that there was a bias to the left in the steady states caused by modeling errors as can be seen in Fig.9(c). Moreover, the landing impact at the first step can cause lateral disturbances.

In the bottom part of Fig.8, the peak force was 116 N and the impulse was 36.0 Ns ((iv) in Fig.9). The robot first modified the landing position/timing and angular momentum for the x component, and then tried to modify the landing position/timing and angular momentum for both the x and y components at the second step. However, the robot detected that the modification could not stabilize walking even after stepping more toward the left, and then performed the fall motion at approximately 30.0 s. Owing to detection before completely falling, there was enough time to change the motion; therefore, the robot avoided damage caused by strong contact with the ground. As a result, the robot was able to stand up again soon after falling, without help from humans.

C. Side–push recovery

Fig.11 shows the snapshots of the side–push recovery experiment. The robot was pushed to the right while the right foot was supporting at a nominal step cycle of 1.2 s. Fig.12(a) shows the external force and Fig.12(b) shows the modified landing timing. The peak force was 43 N and the impulse was 15.6 Ns.

The robot extended the step time for moving the CP close to the left foot and then modified the landing position and timing at the second step. It was verified that the step time

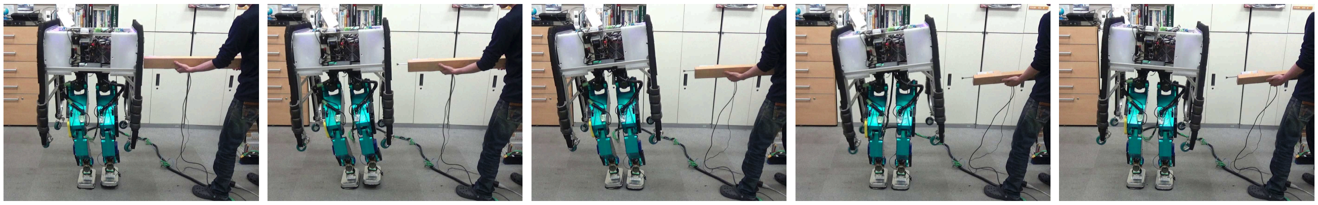
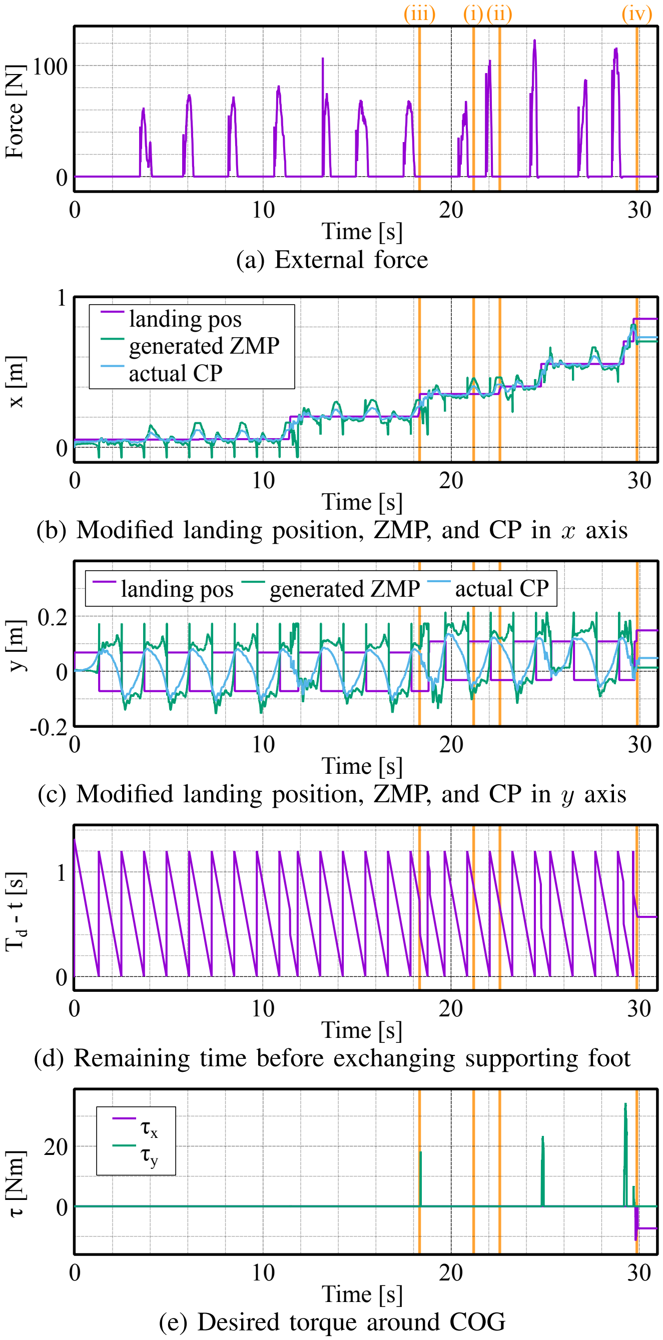


Fig. 11. Experiment for side–push recovery while walking on the spot. The peak force was 43 N and impulse was 15.6 Ns. The robot was pushed to the right during the right support phase. The step time was extended at the first step and shortened at the second step.



(a) External force

(b) Modified landing position, ZMP, and CP in x axis

(c) Modified landing position, ZMP, and CP in y axis

(d) Remaining time before exchanging supporting foot

(e) Desired torque around COG

Fig. 9. Experimental data of the forward push recovery experiment. (i) : Modified the ZMP within the support polygon. (ii) : Modified the landing position. (iii) : Modified the landing position/timing and angular momentum. (iv) : Detected falling.

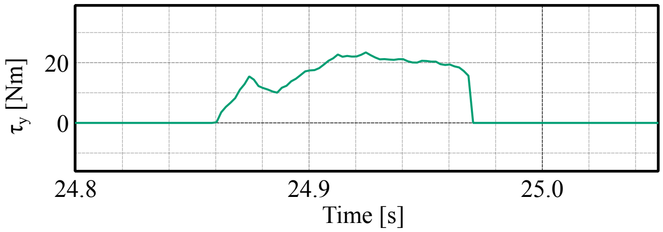
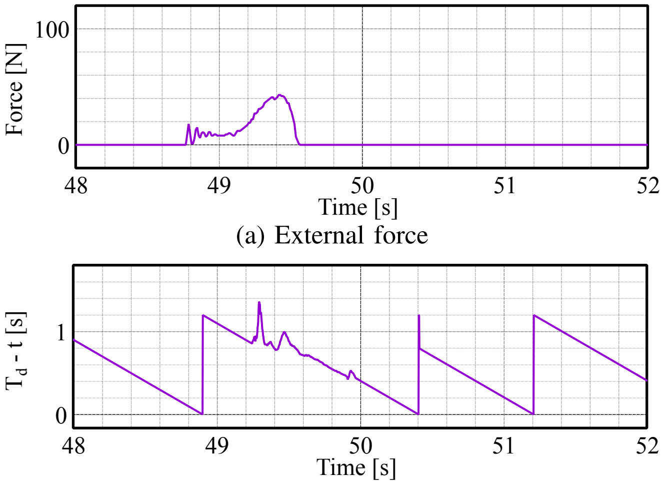


Fig. 10. Zoomed view of Fig.9(e)



(a) External force

(b) Remaining time before exchanging supporting foot

Fig. 12. Experimental data of the side–push recovery experiment

was shortened or extended depending on the direction and timing of external forces.

V. Conclusion

In this paper, we proposed a walking pattern generation method including the landing position and timing modification. We showed that the step time was extended or shortened depending on the magnitude and direction of disturbances by separating the Capturability into sagittal and lateral components. Furthermore, we extended this method to control the angular momentum, which enabled the robot to withstand larger disturbances. The robot could prevent breakdown by detecting possible falling and performed knee bending motions to suppress damage. We confirmed the effectiveness of the proposed methods via experiments in which the biped robot with a protector was pushed forward and sideways while walking without a tether.

REFERENCES

[1] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi. Biped walking stabilization based on linear inverted pendulum tracking. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496, Oct. 2010.

[2] A. Pajon, S. Caron, G. D. Magistri, S. Miossec, and A. Kheddar. Walking on gravel with soft soles using linear inverted pendulum tracking and reaction force distribution. In *Proceedings of the 2017 IEEE-RAS International Conference on Humanoid Robots*, pp. 432–437, 2017.

[3] J. Englsberger and C. Ott. Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking. In *Proceedings of the 2012 IEEE-RAS International Conference on Humanoid Robots*, pp. 183–189, 2012.

[4] K. Hashimoto, H. Kang, M. Nakamura, E. Falotico, H. Lim, A. Takanishi, C. Laschi, P. Dario, and A. Berthoz. Realization of biped walking on soft ground with stabilization control based on gait analysis. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2064–2069, Oct. 2012.

[5] S. Feng, X Xinjilefu, C. G. Atkeson, and J. Kim. Robust dynamic walking using online foot step optimization. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5373–5378, 2016.

[6] T. Kamioka, H. Kaneko, T. Takenaka, and T. Yoshiike. Simultaneous optimization of zmp and footsteps based on the analytical solution of divergent component of motion. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, pp. 1763–1770, 2018.

[7] H. Jeong, O. Sim, H. Bae, K. Lee, J. Oh, and J. Oh. Biped walking stabilization based on foot placement control using capture point feedback. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5263–5269, 2017.

[8] K. Nishiwaki and S. Kagami. Strategies for adjusting the zmp reference trajectory for maintaining balance in humanoid walking. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 4230–4236, 2010.

[9] G. Wiedebach, S. Bertrand, T. Wu, L. Fiorio, S. McCrory, R. Griffin, F. Nori, and J. Pratt. Walking on partial footholds including line contacts with the humanoid robot atlas. In *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, pp. 1312–1319, 2016.

[10] M. A. Hopkins, D. W. Hong, and A. Leonessa. Compliant locomotion using whole-body control and divergent component of motion tracking. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*, pp. 5726–5733, 2015.

[11] B. Stephens. Humanoid push recovery. In *Proceedings of the 2007 IEEE-RAS International Conference on Humanoid Robots*, pp. 589–595, 2007.

[12] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *Proceedings of the 2006 IEEE-RAS International Conference on Humanoid Robots*, pp. 200–207, Dec. 2006.

[13] S. Kajita, O. Matsumoto, and M. Saigo. Real-time 3d walking pattern generation for a biped robot with telescopic legs. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2299–2306, 2001.

[14] T. Koolen, T. D. Boer, J. Rebula, A. Goswami, and J. Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The Internationaltional Journal of Robotics Research*, Vol. 31, No. 9, pp. 1094–1113, 2012.

[15] T. Sugihara and T. Yamamoto. Foot-guided agile control of a biped robot through zmp manipulation. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4546–4551, 2017.

[16] T. Yamamoto and T. Sugihara. Falling prevention of a biped robot by landing point correction based on capture region (in Japanese). In *Proceedings of the 18th SICE System Integration Division Annual Conference*, pp. 3B6–05, 2017.

[17] V. Samy and A. Kheddar. Falls control using posture reshaping and active compliance. In *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*, pp. 908–913, 2015.

[18] J. Lee, W. Choi, D. Kanoulas, R. Subburaman, D. G. Caldwell, and N. G. Tsagarakis. An active compliant impact protection system for humanoids: Application to walk-man hands. In *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, pp. 778–785, 2016.

[19] Y. Kakiuchi, M. Kamon, N. Shimomura, S. Yukizaki, N. Takasugi, S. Nozawa, K. Okada, and M. Inaba. Development of life-sized humanoid robot platform with robustness for falling down, long time working and error occurrence. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 689–696, 2017.

[20] M. B. Popovic, A. Goswami, and H. Herr. Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *The Internationaltional Journal of Robotics Research*, Vol. 24, No. 12, pp. 1013–1032, 2005.

[21] T. Sugihara. Solvability-unconcerned inverse kinematics by the levenberg–marquardt method. *IEEE Transactions on Robotics*, Vol. 27, No. 5, pp. 984–991, 2011.

[22] B. Hoff and M. A. Arbib. Models of trajectory formation and temporal interaction of reach and grasp. *Journal of Motor Behavior*, Vol. 25, No. 3, pp. 175–192, 1993.

[23] S. Nakaoka. Choreonoid: Extensible virtual robot environment built on an integrated gui framework. In *Proceedings of the 2012 IEEE/SICE International Symposium*, pp. 79–85, 2012.

[24] K. Kojima, T. Karasawa, T. Kozuki, E. Kuroiwa, S. Yukizaki, S. Iwaishi, T. Ishikawa, R. Koyama, S. Noda, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba. Development of Life-Sized High-Power Humanoid Robot JAXON for Real-World Use. In *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*, pp. 838–343, Nov. 2015.