Journal of Intelligent & Robotic Systems (2019) 96:345–361
https://doi.org/10.1007/s10846-018-0965-7

Stability Control for Dynamic Walking of Bipedal Robot with Real-time Capture Point Trajectory Optimization

In-Seok Kim[1] · Young-Joong Han[1] · Young-Dae Hong[1]

Received: 30 January 2018 / Accepted: 29 November 2018 / Published online: 12 January 2019

Abstract

This paper proposes a stabilization method for dynamic walking of a bipedal robot with real-time optimization of capture point trajectories. We used the capture point trajectories to generate the control input, which is the desired zero moment point (ZMP) with a sliding-mode ZMP controller to follow the desired ZMP. This method enables the robot to implement various dynamic walking commands, such as forward stride, lateral stride, walking direction, single support time, and double support time. We also adopted enhanced dynamics with the three mass linear inverted pendulum model (3M-LIPM). First, the compensated ZMP is calculated by both walking commands and kinematic configuration of the robot in closed form. Then, the walking pattern is obtained by using initial and boundary conditions of the 3M-LIPM, which satisfies the walking commands. The capture point (CP) trajectory is optimized in real time to control the walking stability and a capture point tracking controller is used for tracking the optimized CP trajectory, which generates an optimal control input that is near the center of the support polygon. The performance of the proposed stabilization method was verified by a dynamics simulator, Webots, and comparison with the original capture point controller-based walking algorithm is presented.

Keywords Three mass inverted pendulum model · Walking pattern generation · Capture point dynamics-based walking · Divergent component of motion · Real-time walking optimization

1 Introduction

HUMANOID robots have attracted many researchers in the field of robotics for more than 20 years as human-like flexible mobility is expected to be possible with a bipedal mechanism. Walking stability has been considered one of the important aspects of bipedal walking as it is related to the flexible mobility of humanoid robots. As a stability criterion of walking, zero moment point (ZMP) has been used as it is simple to measure and calculate [1]. The ZMP is the point on the foot at which horizontal components of the resultant moment generated by active forces and moments acting on humanoid robot are equal to zero. If the ZMP is on the edge in the support polygon of the supporting foot, the robot starts tilting and losing its stability. To keep the ZMP stable while the robot is walking, the walking pattern generation method based on the linear inverted pendulum model (LIPM) is widely used. LIPM-based walking pattern generation makes it possible to implement dynamic walking of bipedal robot and it is easy to analyze the dynamic behavior of the model. The LIPM dynamics can be decoupled in sagittal and lateral planes and this property of the model is useful for generating walking patterns [2–4]. Many variations of this approach have been studied [5–10]. One of the drawbacks of this approach is the lack of accuracy of the dynamics model of the robot, which can lead to walking instability. Many efforts have been made to overcome this problem by adding more masses, like in the gravity compensated inverted pendulum model (GCIPM)

✉ Young-Dae Hong
ydhong@ajou.ac.kr

In-Seok Kim
epickis89@gmail.com

Young-Joong Han
besttopjoong1@ajou.ac.kr

[1] Department of Electrical and Computer Engineering, Ajou University, Suwon 443-749, Korea

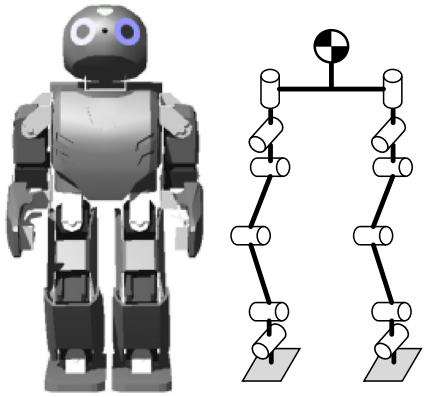


Fig. 1 The robot DARwIn-OP configuration (left) and the kinematic configuration of the lower part of the robot needed to implement walking (right)

[11] and the three mass inverted pendulum model [12]. There have also been stabilization approaches with LIPM-based walking pattern generation to enhance the walking stability [13–15].

One of the widely used walking stability controls is the capture point (CP) based walking method. The CP is the point the robot has to step to in order to stop at a particular point [16, 17]. Some researchers used this property to derive walking controllers [18–22]. The walking algorithms use the CP trajectory to generate a control input which is used to get walking patterns. This type of walking algorithm uses the property that the control input and the center of mass (CM) trajectory is coupled in cascade form. The LIPM

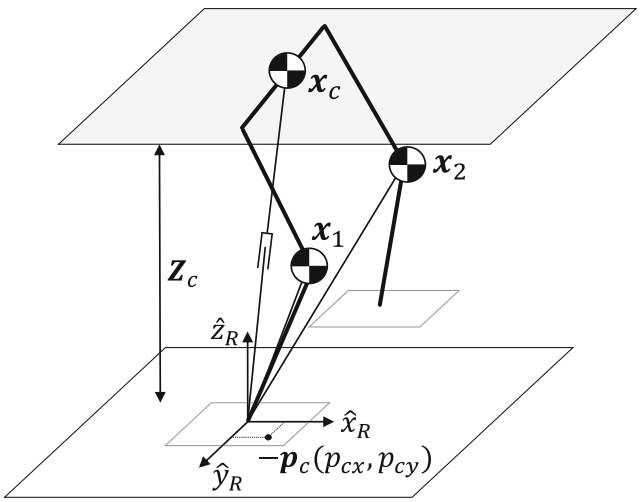


Fig. 2 The three mass inverted pendulum model depicted in 3D space. x_c is the CM of the robot, x_1 is the position of the mass attached on the right knee, and x_2 is the position of the mass attached on the left knee

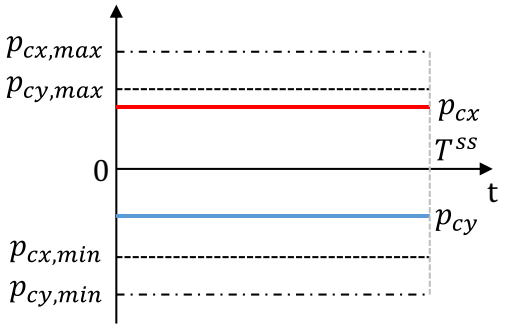


Fig. 3 The cZMP to generate the walking pattern with the 3M-LIPM. p_{cx} is the sagittal cZMP and p_{cy} is the lateral cZMP

dynamics can be decomposed into stable and unstable terms that are closely related to the CP [23, 24]. The walking stability can be improved, and the planning process becomes easier with CP control methods.

This paper proposes a stabilization method for the robot to implement dynamic walking commands with real-time CP trajectory optimization to generate an optimal control input that is the desired ZMP (dZMP) given to the robot to follow the optimized CP trajectory. To work in dynamic environments that have many dynamic moving obstacles, the humanoid robot must implement dynamic walking commands to avoid collisions. Widely used conventional CP controllers have a drawback that the control input, dZMP, can be generated at the edge of the support polygon which can tilt the robot and make the robot fall down, and this makes dynamic walking commands be infeasible. The stabilization method proposed in this paper minimizes this problem and makes the optimal control input, dZMP, near the center of the support polygon. This makes it possible for the robot to perform various dynamic walking commands composed of wide frontal and lateral strides with short single support and double support times. The wide range of the walking commands can be assured to be feasible by using proposed method.

This paper is organized as follows: Section 2 describes the capture point trajectory optimization approach to implement dynamic walking of the bipedal robot. First, the kinematic analysis of the robot is described. Then, the walking pattern generation method with the three mass inverted pendulum model (3M-LIPM) is derived and real-time optimization of capture point trajectory to generate optimal control input based on particle swarm optimization (PSO) is derived. Finally, the ZMP controller using a sliding-mode controller to perform control is described. Section 3 shows simulation results of the proposed methods using a dynamics simulator, Webots, with comparisons to

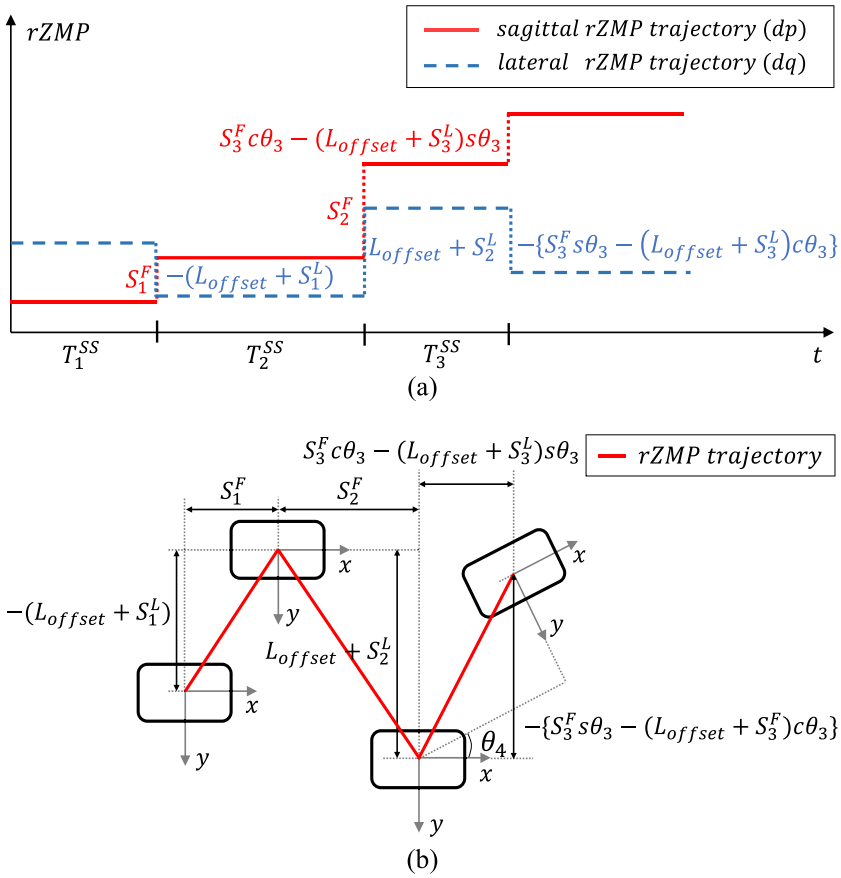


Fig. 4 **a** rZMP trajectory in sagittal and lateral directions. **b** rZMP trajectory on the coronal plane

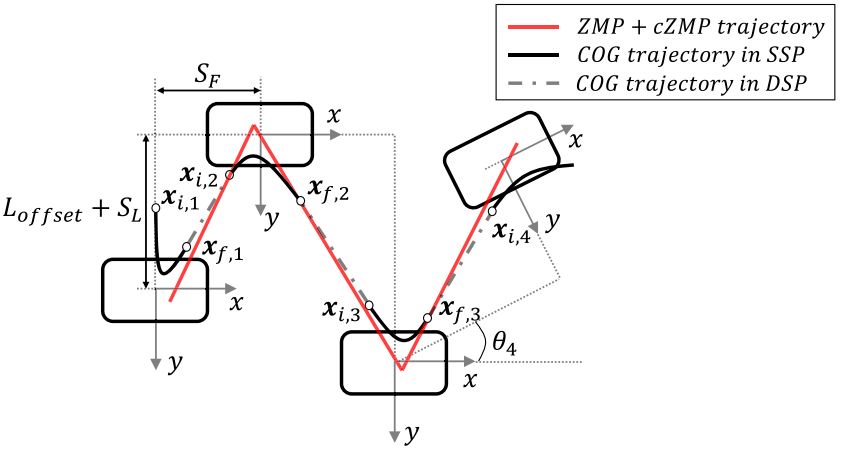


Fig. 5 CM trajectory in the SSP and the DSP

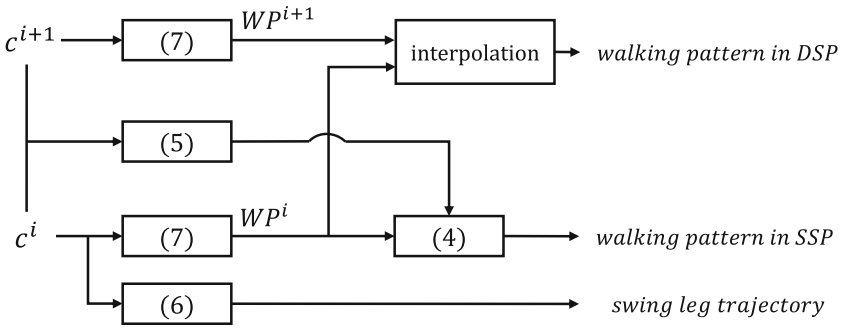


Fig. 6 A flow chart of walking pattern generation using the 3M-LIPM

Table 1 PSO optimization variables

Variables	Initial Position	Initial Velocity
p_x	(−0.045, 0.045)	(−0.05, 0.05)
p_y	(−0.045,0.045)	(−0.04, 0.04)
$\xi_{0,x}$	(−0.1, 0.1)	(−0.05, 0.05)
$\xi_{0,y}$	(−0.155,0.155)	(−0.04, 0.04)

CP tracking controller-based walking. Section 4 concludes the paper with possibilities and future work of the proposed approach.

2 Capture Point Trajectory Optimization-based Stability Control for Dynamic Bipedal Walking

The proposed stabilization method for dynamic walking of bipedal robot is composed of three stages. First, the walking pattern is generated using the 3M-LIPM which satisfies the walking commands with walking primitives. Secondly, the CP trajectory is optimized in real-time using a PSO algorithm to make the optimal control input, dZMP. Then, a ZMP controller modifies the CM trajectory to control the ZMP and follow the optimized CP trajectory in a position controlled robot platform DARwIn-OP.

2.1 Robot Kinematics Analysis

The DARwIn-OP, shown in Fig. 1, was used to verify the performance of the proposed stabilization method. The kinematics are solved by homogeneous transformation matrices from the base (supporting foot) to the CM and to the end of the link (swing foot) as follows:

$$R_1^{COG} = \prod_{i=1}^{5} R_i^{i+1}, R_1^{end} = \prod_{i=1}^{11} R_i^{i+1} \tag{1}$$

A 6x1 vector composed of positions and orientations is obtained by Eq. 1. The inverse kinematics are solved by an iterative method using Jacobian transpose matrices with roll-pitch-yaw (RPY) conventions as follows:

$$\dot{x}_1^{COG} = \mathbf{J}_1^{COG}\dot{\theta}_{6\times1}, \dot{x}_1^{end} = \mathbf{J}_1^{end}\dot{\theta}_{12\times1} \tag{2}$$

where x_1^{COG} is a 6x1 matrix, $\theta_{6\times1}$ is a 6×1 matrix and $\theta_{12\times1}$ is a 12×1 matrix composed of joint angles, J_1^{COG} is a 6×6 matrix and J_1^{end} is a 6×12 matrix. Inverse kinematics are easily analyzed by using the transpose of Eq. 2, which

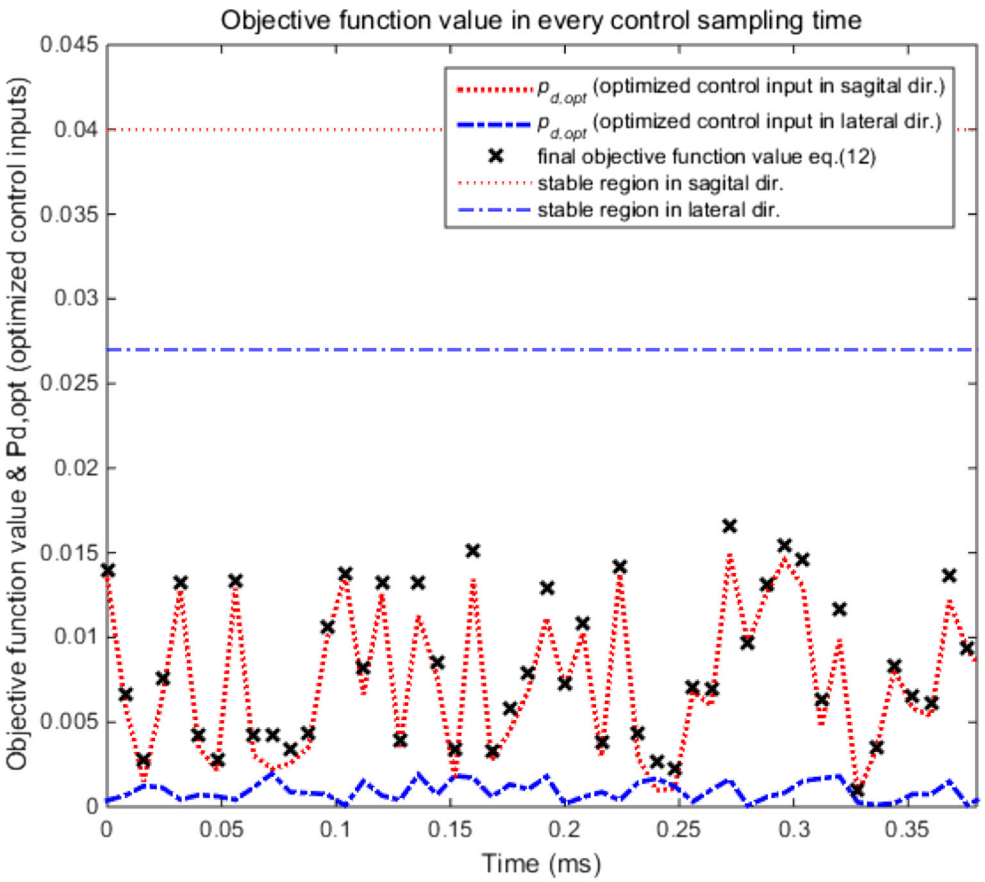


Fig. 7 Objective function values after PSO optimization with every control sampling time in one of the walking commands

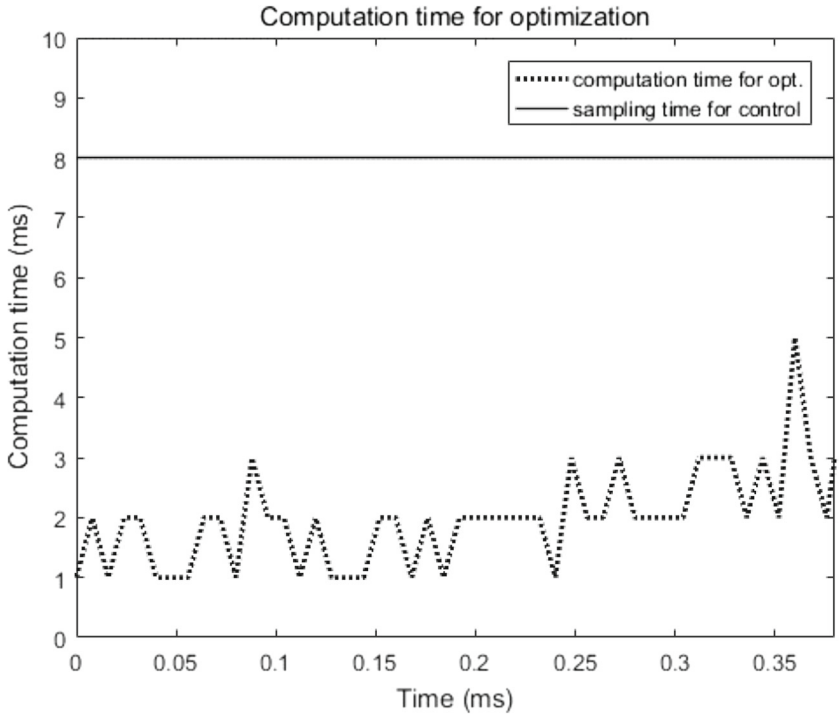


Fig. 8 An example of computation time for the PSO optimization with every control sampling time in one of the walking commands

is the pseudo-inverse iterative approach to solving inverse kinematics.

2.2 Walking Pattern Generation with a Three Mass Model

A walking pattern is generated with the three mass inverted pendulum model as shown in Fig. 2. Unlike widely used walking pattern generation based on multiple mass models, we used the original LIPM dynamics which is single mass model to represent dynamics of the robot and adopted a compensated ZMP that is derived from two masses attached on the right and left knees as it can be used in real-time pattern generation due to computational simplicity. The dynamics of the LIPM are derived by using the ZMP equation as follows [2, 3]:

$$\ddot{x}_c = \frac{g}{Z_c}(x_c - p) \tag{3}$$

where $x_\mathbf{c}$ is the position of the CM with respect to support origin, p is the ZMP vector with respect to the support polygon's origin, Z_c is the height of the CM, and g is the gravity constant. As we can decouple sagittal and lateral dynamics with LIPM, x_c in Eq. 3 is either sagittal or lateral position of CM and p in Eq. 3 is also either sagittal or lateral ZMP. The explicit solution of the differential equation (Eq. 3) decoupled in sagittal and lateral directions in the time domain is easily obtained by using the Laplace transform as follows:

$$\begin{bmatrix} x_c(t) \\ \dot{x_c}(t) \end{bmatrix} = \begin{bmatrix} \cosh\left(\frac{t}{T_c}\right) & T_c \sinh\left(\frac{t}{T_c}\right) \\ \frac{1}{T_c}\sinh\left(\frac{t}{T_c}\right) & \cosh\left(\frac{t}{T_c}\right) \end{bmatrix} \begin{bmatrix} x_0 \\ \dot{x_0} \end{bmatrix} + \begin{bmatrix} 1 - \cosh\left(\frac{t}{T_c}\right) \\ -\frac{1}{T_c}\sinh\left(\frac{t}{T_c}\right) \end{bmatrix} p \tag{4}$$

where T_c is $\sqrt{g/Z_c}$. Conventional walking pattern generation methods use the CM trajectory obtained from Eq. 3 with $p = 0$, which means that the resulting ZMP is assured to be at the center of the support polygon when the robot tracks the generated CM trajectory. Although it is simple and reasonable to use a walking pattern generated with $p =$

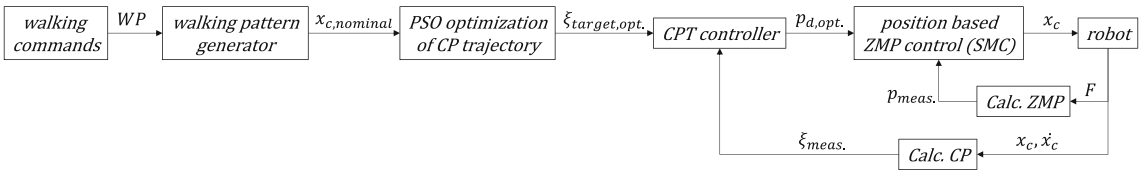


Fig. 9 Overall real-time capture point trajectory optimization based stability control method for dynamic walking of bipedal robot

0, modeling error is sometimes too significant, especially when drastic walking commands are needed. Therefore, in order for the robot to perform stable walking we used the concept of the compensated ZMP (cZMP) to compensate this modeling errors by adding two masses on both sides of the knees. The idea is simple. First, the walking pattern satisfying the walking command is obtained using the LIPM dynamics. Then the trajectory of the additional masses is obtained by similar triangles formulated to a proportional expression when the CM tracks the previously obtained walking pattern that satisfies the walking command. Then, the ZMP acting on the support foot due to the motion of the additional masses can be computed by the ZMP equation as follows:

$$p_c = \frac{\sum_{i=1}^{2} m_i \{(\ddot{z}_i + g) x_i - (z_i - p) \ddot{x}_i\}}{\sum_{i=1}^{2} m_i (\ddot{z}_i + g)} \quad (5)$$

where p_c is a ZMP caused by motion of the mass and x_i, z_i are position components of the mass. The mean of the cZMP is obtained by integrating Eq. 5 from *0* to single support time, T^{SS} and dividing it by T^{SS}. The cZMP can compensate the modeling error of the LIPM. As the ZMP caused by the additional masses are cZMP, if we put the opposite of cZMP, −cZMP into Eq. 4, we can cancel this effects out. The resulting scalar values $-p_{cx}$ or $-p_{cy}$ are used to generate a walking pattern as an input p to Eq. 4 to compensate the effects of additional masses. Therefore, the walking pattern generator generates CM trajectory that compensates the effects of supporting leg and swing leg by substitution of the opposite sign of cZMP. The cZMP has a form of scaled step function from $t = 0$ to $t = T^{SS}$ as shown in Fig. 3. There are possible regions for sagittal and lateral cZMPs that are the sizes of the foot in the sagittal $(p_{cx,min}, p_{cx,max})$ and lateral $(p_{cy,min}, p_{cy,max})$ directions. This method is useful as it is simple to calculate, which makes it possible to generate real-time walking patterns and as many additional masses as possible can be attached if needed.

The walking pattern generation method proposed in this paper can make the bipedal robot perform various walking motions following walking commands which are sent to the robot by some kind of footstep planners or path planners, which are a layer above pattern generation and the control layer. The walking command is defined as follows:

$$c_i = [S_i^F, S_i^L, \theta_i, T_i^{SS}, T_i^{DS}]^T$$

where the subscript is the sequence number of the walking command, S^F is sagittal step length (frontal stride), S^L is lateral step length (lateral stride), θ is a direction of walking (walking angle), T^{SS} is single support time during the single support phase (SSP), and T^{DS} is double support time during the double support phase (DSP). To generate the walking pattern satisfying the walking command, a reference ZMP (rZMP) needs to be planned and this rZMP is used to calculate the next foot position, which is the position of the swing foot when it lands on the ground with respect to the supporting foot. This rZMP is also used to generate the reference walking pattern and this is used to calculate the cZMP generated by the motion of reference walking pattern. The rZMP is planned in world coordinates as follows:

$$\begin{bmatrix} dp_{i+1} \\ dq_{i+1} \end{bmatrix} = \begin{bmatrix} dp_i \\ dq_i \end{bmatrix} + \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} S_i^F \\ (-1)^i (S_i^L + L_{offset}) \end{bmatrix} \quad (6)$$

where dp is sagittal rZMP, dq is lateral rZMP, and L_{offset} is a constant the same as the size of the pelvis of the robot. An example of planning the rZMP is shown in Fig. 4. The walking primitives (WP) are defined as the initial and final positions of CM in the SSP as follows:

$$WP_i = \left[x_i^{cog,i}, y_i^{cog,i}, x_i^{cog,f}, y_i^{cog,f}\right]^T$$

where $x_i^{cog,i}, y_i^{cog,i}$ is the initial position of the CM and $x_i^{cog,f}, y_i^{cog,f}$ is the final position of the CM. The final position of the CM in the SSP becomes the initial position of the CM in the DSP and the final position of the CM in the DSP with WP_i becomes the initial position of the CM with WP_{i+1}. WP is a boundary condition for the CM and is calculated as follows:

$$\begin{bmatrix} x_i^{cog,i} \\ y_i^{cog,i} \end{bmatrix} = \mathbf{diag}(k_F, k_L) \begin{bmatrix} -S_{i-1}^F \\ (-1)^i (S_{i-1}^L + L_{offset}) \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} x_i^{cog,f} \\ y_i^{cog,f} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \mathbf{diag}(k_F, k_L) \begin{bmatrix} S_i^F \\ (-1)^i (S_i^L + L_{offset}) \end{bmatrix}$$

where k_F is the sagittal primitive factor and k_L is the lateral primitive factor. Both of these primitive factors range from 0 to 1/2. If the primitive factors are 1/2, there is no DSP in the walking pattern. The CM trajectory in the DSP is affected by primitive factors and it gets longer if the primitive factor becomes close to zero. We used 1/4 for k_F and k_L. Figure 5 shows the CM trajectory in both the SSP and the DSP when the primitive factors are 1/4. In the DSP, the CM trajectory is obtained by linear interpolation that satisfies the boundary condition defined by the WP. In the SSP, the walking pattern satisfying Eq. 7 is obtained by solving the initial velocity of the CM in Eq. 4 with mean of Eq. 5 as p. The trajectories for the swing leg is obtained by using Cycloid function to minimize the impact of landing foot. The overall process for generating the walking pattern is depicted in Fig. 6. In the view point of state transition, the walking pattern is generated as shown in Fig. 6. It changes to the swing leg flight off state and becomes SSP state then

changes to swing leg landing state, and then DSP state. In the end of DSP, the process repeats and the robot walks the next step.

2.3 Capture Point Trajectory Optimization

The CP is the point where the robot has to land its swing leg to make it stop at a point. The CP is easily derived by decoupling the LIPM dynamics and the robot can walk more stably by tracking the CP trajectory with a CP tracking (CPT) controller. There are two types of advantage of using a CP for walking control. First, the walking pattern of the CM is obtained without considering WP and the boundary conditions as the CM trajectory satisfying the predesigned dZMP is automatically obtained by the CPT controller. Second, the unstable component of motion, which is the so-called divergent component of motion, can be suppressed while walking. The CP dynamics are defined as follows [18, 19]:

$$\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{x}}_c + \frac{\ddot{\boldsymbol{x}}_c}{w} \tag{8}$$

where ξ is the CP and $w = 1/T_c$. The solution of Eq. 8 is as follows [19, 20]:

$$\boldsymbol{\xi}(t) = e^{wt}\boldsymbol{\xi}_0 + (1 - e^{wt})\boldsymbol{p} \tag{9}$$

$$\boldsymbol{\xi}_0 = \boldsymbol{x}_{c,0} + \frac{\dot{\boldsymbol{x}}_{c,0}}{w}$$

where $\boldsymbol{x}_{c,0}$ is the nominal initial CM position and $\dot{\boldsymbol{x}}_{c,0}$ is the nominal initial CM velocity. The nominal CP trajectory is planned as follows:

$$\begin{bmatrix} \xi_{d,x} \\ \xi_{d,y} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} e^{wt}\xi_{0,x} + \left(1 - e^{wt}\right) p_x \\ e^{wt}\xi_{0,y} + \left(1 - e^{wt}\right) p_y \end{bmatrix} \tag{10}$$

where $\boldsymbol{\xi}_d$ is the desired nominal CP trajectory to be planned, $\boldsymbol{\xi_0}$ is the initial CP defined by $\boldsymbol{x}_{c,0} + \dot{\boldsymbol{x}}_{c,0}/w$, and $\boldsymbol{p}$ is the initial ZMP. We can derive a control law to follow the desired CP trajectory as follows [19]:

$$\mathbf{p}_d = \frac{\boldsymbol{\xi}_{ref} - e^{wdT}\boldsymbol{\xi}}{1 - e^{wdT}} \tag{11}$$

where $\boldsymbol{\xi}_{ref}$ is the reference CP trajectory and it is obtained by changing wt to $w(t + dT)$ in Eq. 10, and dT is control sampling time. To track the reference CP trajectory Eq. 11 is used as a control input to the CPT controller [19]. This method has a drawback that the dZMP obtained from Eq. 11 can be outside of the support polygon and some kinds of projection method was proposed to deal with this issue [20]. We solved this problem and generated an optimal dZMP by optimizing the CP trajectory, which makes the robot walk more stably with the dZMP being near the center of the support polygon. This can be interpreted as optimizing the CP offset by adjusting the dZMP and initial CP to match the measured CP to the real CP obtained from the LIPM dynamics. Furthermore, there was no rule to make a reasonable reference to the CP trajectory in the conventional CPT controller, but it was heuristically determined using some offset from the center of the footsteps, which is the predesigned center of the dZMP region, or by using a backwards iteration to set the CP offset [20].

With an optimization approach we can deal with the uncertainties in this problem and obtain an optimal CP offset to generate the optimal control input for the CPT controller. In addition, a backwards iteration is no longer needed to set the CP offset as the optimization automatically generates a CP trajectory. PSO is a popular optimization method for solving global optimization problems as it will nearly always assure a global optimum in solution sets and it is very simple to implement the optimization process. The optimization problem with PSO is formulated to minimize the following objective function:

$$f = \frac{\boldsymbol{\xi}_{ref} - e^{wdT}\boldsymbol{\xi}}{1 - e^{wdT}} \cdot \left(C_1\hat{\boldsymbol{x}}_{sup} + C_2\hat{\boldsymbol{y}}_{sup}\right) + penalty \tag{12}$$

where $\hat{\boldsymbol{x}}_{sup}$ and $\hat{\boldsymbol{y}}_{sup}$ are the unit vector of the supporting foot's orientation and the penalty is to ensure that the end of the CP is in the next supporting polygon. The optimization variables are $\boldsymbol{\xi}_0$ and $\boldsymbol{p}$ in the reference CP trajectory, $\boldsymbol{\xi}_{ref}$. We used 300 particles for PSO optimization with randomly selected initial position and velocity of the particle as shown in Table 1. For the stop criterion of the optimization, we decomposed objective function (12) into sagittal and lateral direction with sagittal component smaller than $C_1 = 0.015$ and lateral component smaller than $C_2 = 0.002$. The stop criterion of the optimization is Eq. 12 smaller than 0.017. With these C_1 and C_2, we can assure that the dZMP is generated near the center of the supporting foot because the size of supporting foot of the DARwIn-OP is (-0.04, 0.04) in sagittal direction and (-0.027, 0.027) in lateral direction. With this optimized control input, dZMP ($p_{d,opt}$), CP can track the optimized CP trajectory by the CPT controller. We can see that the optimization is performed well in every control sampling time as shown in Fig. 7. This method makes the robot implement various dynamic walking commands as the control input is feasible even if the walking commands are drastic as the limitation of the control input is enlarged. This optimization process is implemented in the control sampling time and we used 8ms to optimize the CP trajectory. An example of computation time for the optimization is shown in Fig. 8 and we can conclude that PSO optimization finds the solutions within the control sampling time (8ms) with the given parameters in Table 1. We need a ZMP controller for tracking the optimized control input, $p_{d,opt}$ that makes the current CP track the optimized CP trajectory.

Table 2 Dynamic walking simulation cases

Case	Forward stride	Lateral stride	Rotation	Time
Drastic Dynamic Walking	Dynamic	Dynamic	Fixed	Drastic, Dynamic
Rotational Walking	Fixed	Dynamic	Dynamic	Fixed
Fast Walking	Fixed	Dynamic	Fixed	Short, Dynamic

2.4 ZMP Controller

We used a ZMP controller for the position-controlled robot, DARwIn-OP. We adopted the ZMP controller based on sliding mode control (SMC) to minimize model uncertainty with robustness to disturbances. In the position-controlled robot, the ZMP can be controlled as follows [15, 20]:

$$\ddot{\boldsymbol{x}}_d = -K\frac{F_z}{Z_c}\boldsymbol{e} - \lambda\frac{F_z}{Z_c}\text{sign}(\boldsymbol{\sigma}) \tag{13}$$

where K is the state feedback gain, $\boldsymbol{\sigma}$ is the sliding manifold, λ is control gain, and $\mathbf{e}$ is $(\mathbf{p}_d - \mathbf{p})$. This controller uses a dynamics property that relates the CM acceleration to the ZMP. The ZMP has a property of being shifted in the opposite direction to the CM acceleration. The desired CM position for the ZMP control is obtained from Eq. (13) by integration and the robot CM trajectory is modified by adding this value.

2.5 Overall CP Optimization-Based Walking Stability Control

The overall proposed walking stability control method is depicted in Fig. 9. First, the sequence of walking commands c_i is converted to a walking primitive WP_i and then fed to the walking pattern generator. The walking pattern generator calculates the CM trajectory with the dZMP at the center of the support polygon, and then calculates cZMP using the 3M-LIPM method. The generated walking patterns are used as the nominal CM trajectory. Then the reference CP trajectory is planned by using Eq. (10). PSO is applied before generating a control input in the CPT controller in real-time which means that the optimized solution is obtained within a control sampling time, for which we used 8ms. Then the ZMP controller modifies the CM trajectory to track the dZMP, $p_{d,opt}$, which makes the CP follow the optimized CP trajectory. This process is repeated until the current walking command ends. The next step is then optimized and controlled in the same manner. In the viewpoint of control, the optimized control input, $p_{d,opt}$ is ensured to be continuous in every states including the moment of state change from SSP to DSP as the CPT controller generates continuous control input. The stability in SSP is ensured by tracking desired ZMP generated in the center of support polygon by CPT controller. The stability in the moment of state transition from SSP to DSP is ensured by 3M-LIPM model as the CM trajectory is planned ending in the support polygon which is a convex hull of two feet. The stability in DSP is ensured as the overall walking stability control shown in Fig. 9 is used, which makes ZMP follows dZMP in the support polygon. Therefore, the stability in all of the states is ensured with proposed walking stability control method with CPT optimization.

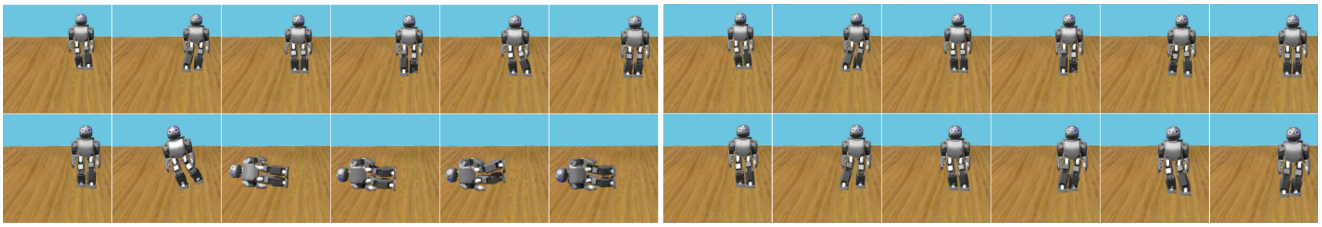
3 Simulations

The performance of the proposed stabilization method is verified through simulation using a dynamics simulator, Webots. We categorized the simulation cases of dynamic walking as shown in Table 2. A drastic dynamic walking case is composed of drastic walking commands, which means that the forward strides, lateral strides, and support time are changed in every step. This drastic dynamic walking is very hard for the robot to implement due to the modeling inaccuracy. The rotational walking case is tested to confirm that a general walking command is possible and that the rotational motion is more stable and feasible with the proposed method. The fast walking case, which is normally hard to implement as the robot loses its stability due to modeling inaccuracy, is tested to confirm that the robot can implement fast walking commands with the proposed method. The conventional CPT controller-based walking cases are compared to the proposed stabilization method for all of the simulation cases.

We first tested drastic dynamic walking commands. The robot should implement various dynamic walking

Table 3 Drastic dynamic walking commands

Walking commands	Forward stride [m]	Lateral stride [m]	Rotation [deg]	SSP Time [s]	DSP Time [s]
c_1	0.04	0.0	0.0	0.6	0.2
c_2	0.06	0.0	0.0	0.7	0.2
c_3	0.065	0.035	0.0	0.55	0.2
c_4	0.06	0.0	0.0	0.45	0.2
c_5	0.0	0.0	0.0	0.5	0.2
c_6	0.05	0.025	0.0	0.35	0.2
c_7	0.025	0.0	0.0	0.5	0.2
c_8	0.05	0.055	0.0	0.35	0.2
c_9	0.07	0.0	0.0	0.4	0.2
c_{10}	0.01	0.0	0.0	0.5	0.2
c_{11}	0.03	0.0	0.0	0.35	0.2
c_{12}	0.07	0.0	0.0	0.75	0.2



(a) (b)

Fig. 10 Implementing drastic dynamic walking c_1 to c_{12} with **a** the conventional CPT controller and **b** the proposed stabilization method

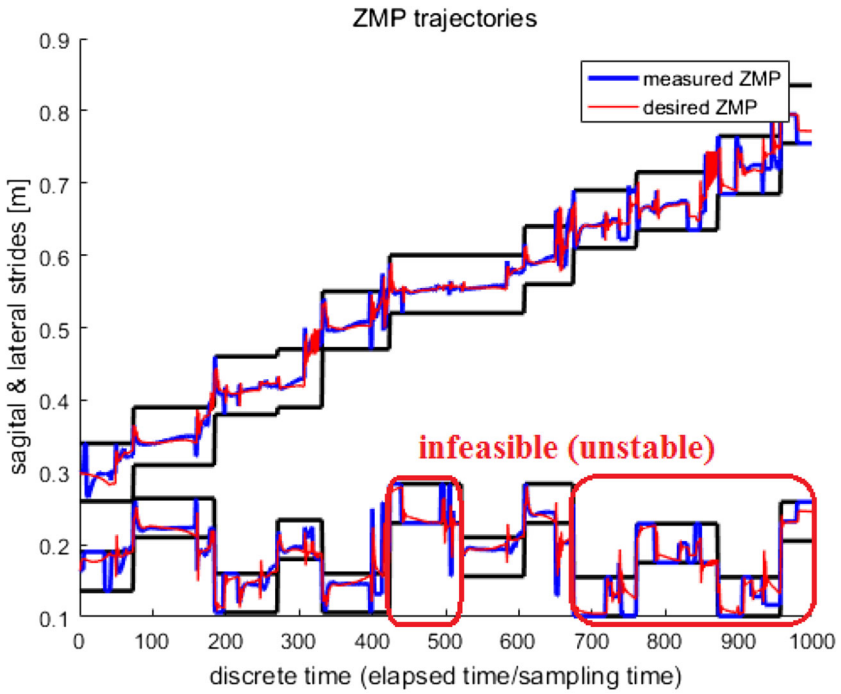


Fig. 11 The measured ZMP in both the sagittal and lateral directions (conventional CPT)

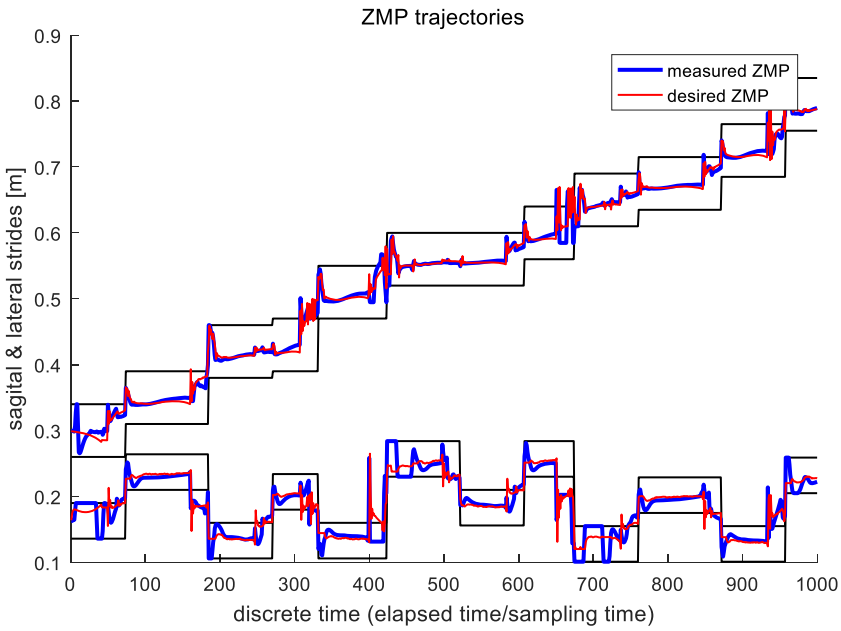


Fig. 12 The measured ZMP in both the sagittal and lateral directions (proposed stabilization method)

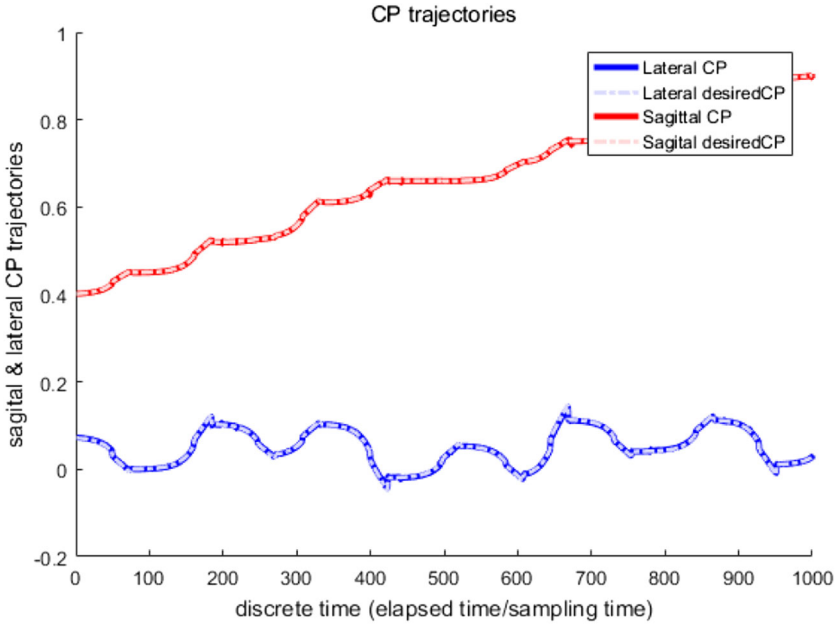


Fig. 13 CP tracking in the sagittal and lateral directions (conventional CPT controller)

commands to cope with dynamic moving obstacles, such as avoiding moving obstacles, stopping suddenly, accelerating suddenly, or a large movement. The proposed stabilization method makes it possible for the robot to perform these tasks as the robot can stably implement drastic dynamic walking commands. The tested walking commands are shown in Table 3. It was noticed that the robot loses its stability with the conventional CPT controller while the proposed method with a CPT controller can implement all of the drastic dynamic walking commands without losing walking stability, as shown in Fig. 10. The conventional CPT controller-based walking is infeasible for c_9 to c_{12}, as shown in Fig. 11, which is represented by the red box, while the proposed stabilization method can implement these dynamic walking commands stably, as shown in Fig. 12. The ZMP is saturated at the edge of the support polygon

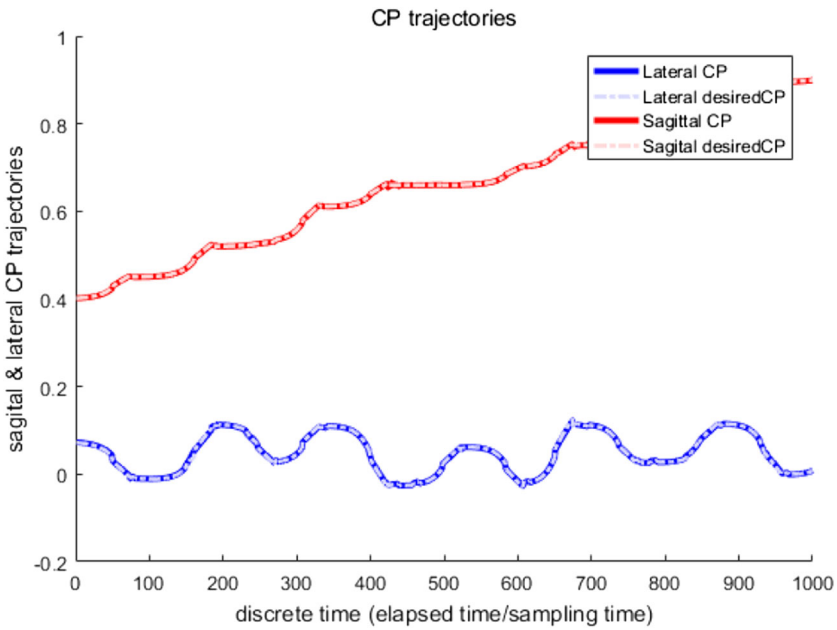


Fig. 14 CP tracking in the sagittal and lateral directions (proposed stabilization method)

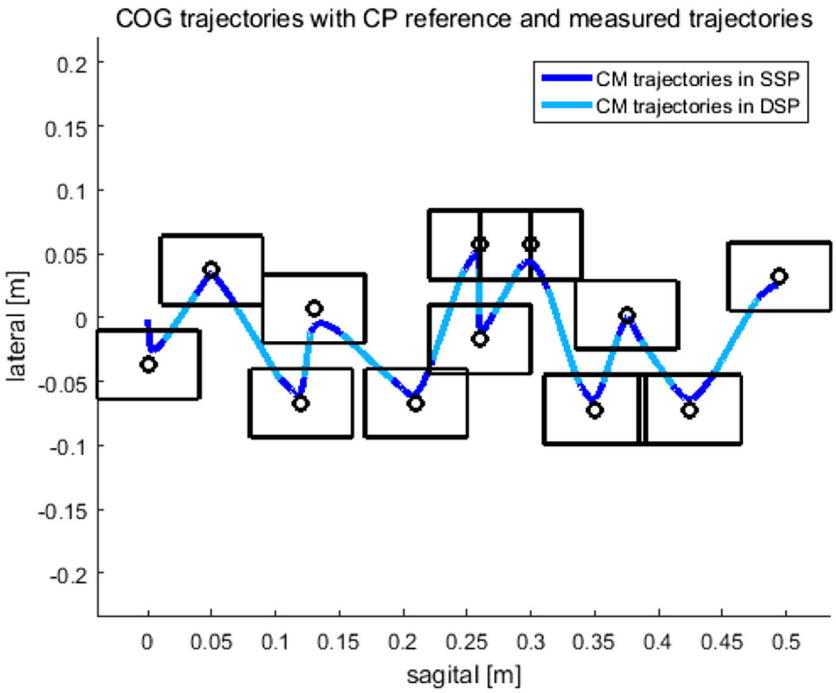


Fig. 15 CM trajectory in the SSP and the DSP (conventional CPT controller)

in CPT controller-based walking and the control input is generated near the edge of the support polygon, which means that it is not feasible to follow the drastic dynamic walking commands. On the other hand, the proposed stabilization method makes the system stable, as shown in Fig. 12. There is no saturation of the control input and the measured ZMP shows that it recovers the walking stability in a short period of time. In addition, both the control input and the measured ZMP are generated nearer the center of the support polygon than in conventional CPT controller-based

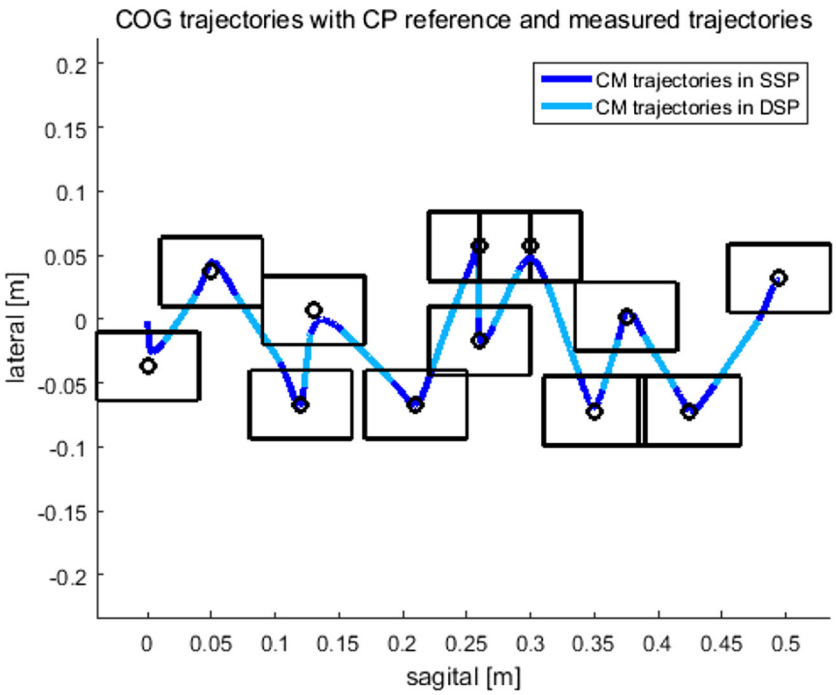


Fig. 16 CM trajectory in the SSP and the DSP (proposed stabilization method)

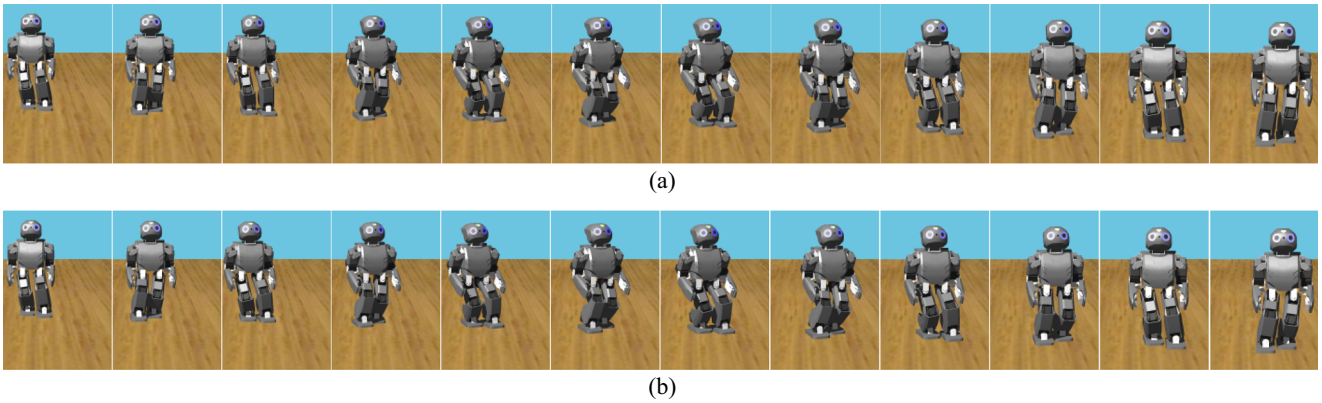
Table 4 Rotational dynamic walking commands

Walking commands	Forward stride [m]	Lateral stride [m]	Rotation [deg]	SSP Time [s]	DSP Time [s]
c_1	0.05	0.0	0.0	0.6	0.2
c_2	0.05	0.0	5.0	0.6	0.2
c_3	0.05	0.0	5.0	0.6	0.2
c_4	0.05	0.0	5.0	0.6	0.2
c_5	0.05	0.0	5.0	0.6	0.2
c_6	0.05	0.0	0.0	0.6	0.2
c_7	0.05	0.0	0.0	0.6	0.2
c_8	0.05	0.0	−5.0	0.6	0.2
c_9	0.05	0.0	−5.0	0.6	0.2
c_{10}	0.05	0.0	−5.0	0.6	0.2
c_{11}	0.05	0.0	−5.0	0.6	0.2
c_{12}	0.05	0.0	0.0	0.6	0.2

walking, which comes from the 3M-LIPM-based walking pattern generation and an optimal control input by CP trajectory optimization. Figures 13 and 14 show the sagittal and lateral CP tracking results by the CPT controller without and with the proposed CP optimization, respectively. The optimization-based CP result scale is shorter than the conventional CP as the control input should be generated near the support polygon. These variations of the CP offset can make the drastic dynamic commands feasible. It can also be seen that the CP ends at the landed foot (the next support foot), which means that CP-based walking still holds. Figures 15 and 16 show the CM trajectory without and with CP optimization, respectively. The proposed optimization-based stabilization method is shown in Fig. 16 and the CM in the SSP shifts more to the support polygon due to the stability control than conventional CPT-based walking. In the proposed stabilization method, the control input is generated optimally near the center of the support polygon and the ZMP controller shifts the CM to the center of the support polygon to follow the control input.

The second dynamic walking simulation case is the dynamic rotational walking case. This rotational command is tested to confirm that the proposed stabilization method is more stably implemented for rotational motions and that general walking commands including rotational motion are feasible. The tested rotational walking commands are shown in Table 4. It can be seen that the robot can implement rotational motions as shown in Fig. 17. Although it is hard to notice that slips occur during walking with conventional CPT controller-based walking, the proposed stabilization method with a CPT controller shows that it does not have slip motions during rotational motions. The ZMP is measured nearer to the center of the support polygon with the proposed method than conventional CPT controller-based walking, as shown in Figs. 18 and 19, and this means that the proposed optimization-based stabilization method is more stable in general cases, including rotational motions.

The third dynamic walking simulation is the fast walking case. Fast walking dynamic commands are tested to confirm that the proposed stabilization method is feasible and more stable in fast walking, which is characterized by short periods of single support and double support time. The tested fast dynamic walking commands are shown in Table 5. It is usually very hard for conventional approaches to make these kinds of commands feasible due to the modeling inaccuracy. However, it was noticed that the robot can stably walk very fast with the proposed stabilization method, whereas it tilts and falls down after losing its



(a)

(b)

Fig. 17 **a** Implementing c_1 to c_{12} with the conventional CPT controller **b** Implementing c_1 to c_{12} with the proposed stabilization method

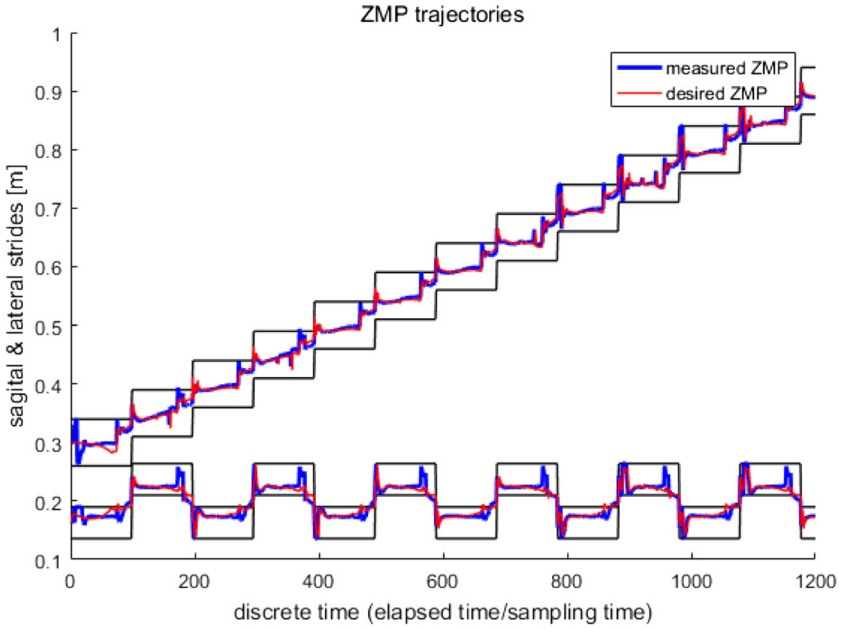


Fig. 18 The measured ZMP in both sagittal and lateral directions (conventional CPT controller)

walking stability with conventional CPT-based walking, as shown in Fig. 20. Conventional CPT controller-based walking is infeasible as the control input is generated near the edge of the support foot, as shown in Fig. 21 The robot starts losing its walking stability from c_9 and starts falling down at c_{11} with the conventional approach. The measured ZMP, shown in Fig. 21, shows that the measured ZMP and the control input are generated at the edge of the support polygon, and the conventional approach cannot recover the stability, and therefore, fails to implement walking commands from c_9 to c_{15}. On the other hand, the proposed stabilization method with a CPT controller makes the system recover its instability in a short period of time, as shown in Fig. 22, with the optimal control input generated near the center of the support polygon.

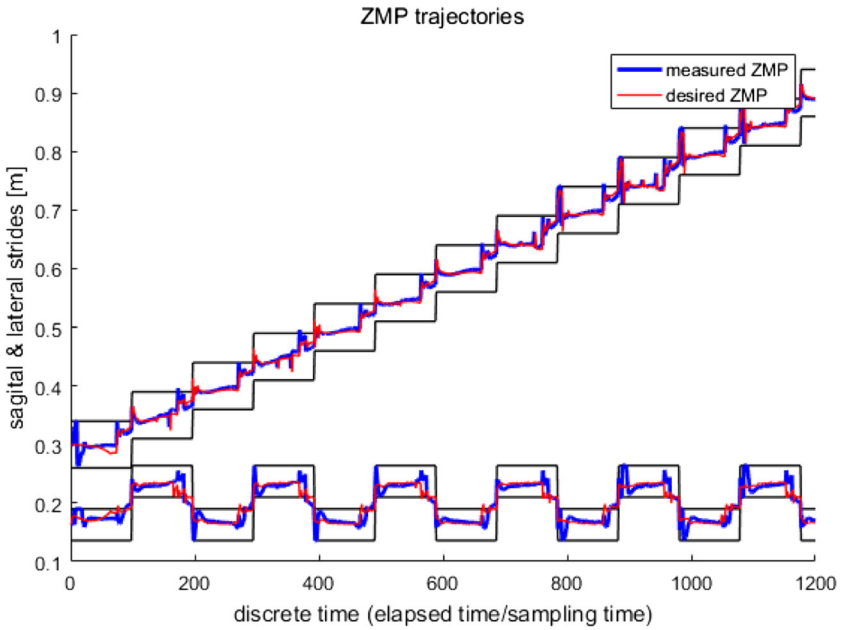


Fig. 19 The measured ZMP in both the sagittal and lateral directions (proposed stabilization method)

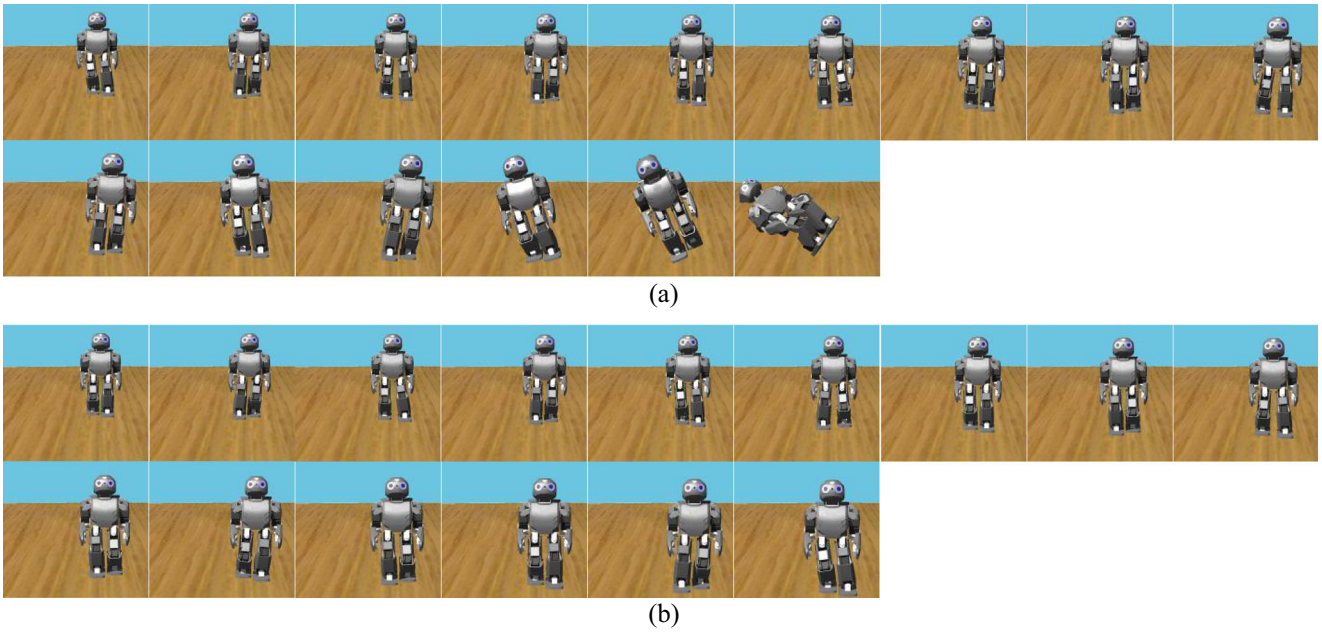


Fig. 20 **a** Implementing c_1 to c_{12} with the conventional CPT controller **b** Implementing c_1 to c_{12} with the proposed stabilization method

In all of the dynamic walking cases tested with the dynamics simulator, Webots, the proposed stabilization method with a CPT controller is confirmed to be much more stable than the conventional CPT controller-based walking algorithm. In addition, the feasibility of the walking commands is assured as the optimization makes the control input feasible, which means that the control input is generated near the center of the support polygon. This

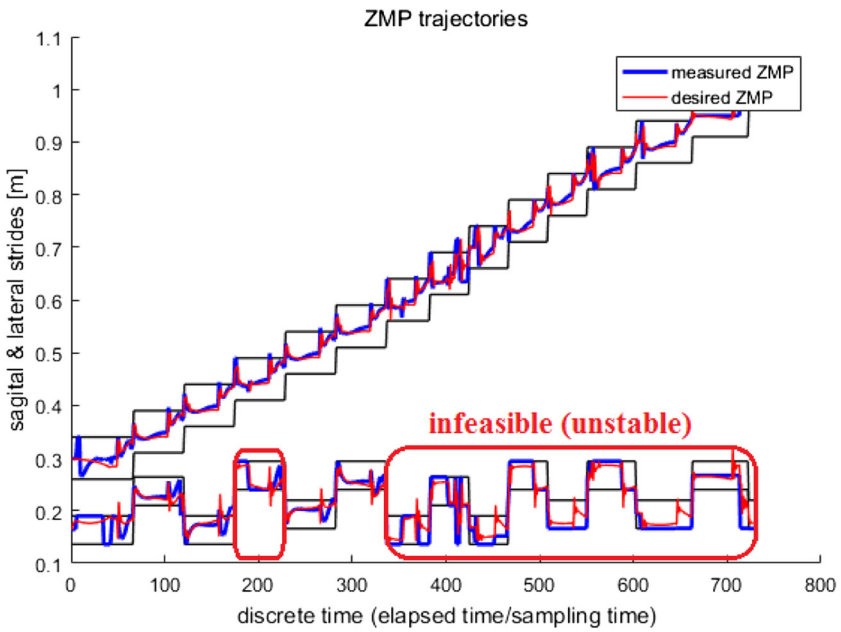


Fig. 21 The measured ZMP in both sagittal and lateral directions (conventional CPT controller)

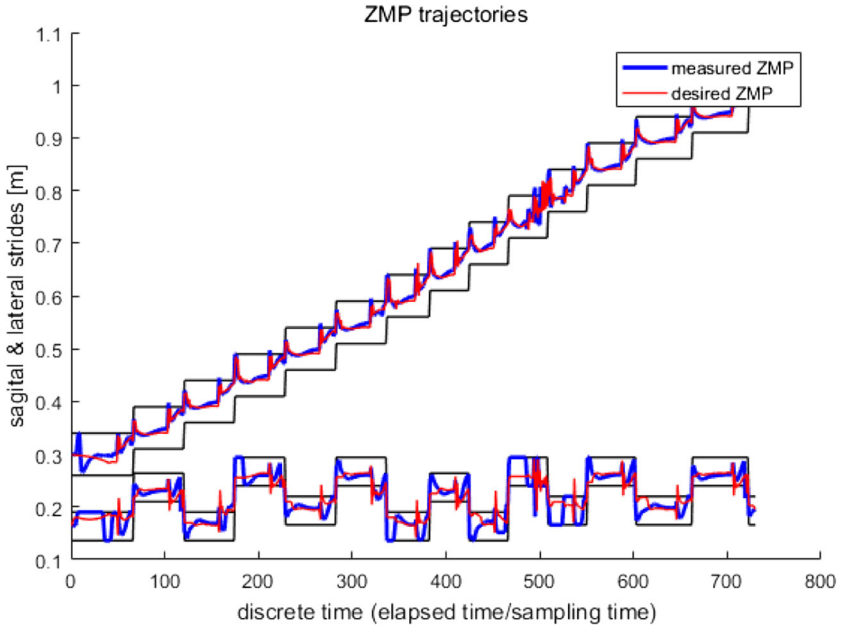


Fig. 22 The measured ZMP in both the sagittal and lateral directions (proposed stabilization method)

optimization makes the robot implement various dynamic walking commands. As it is essential for the robot to implement a large and flexible range of walking commands, we conclude that the simulation results are very promising for dynamic walking from the view point of stability and feasibility.

Table 5 Fast dynamic walking commands

Walking Commands	Forward Stride [m]	Lateral Stride [m]	Rotation [deg]	SSP Time [s]	DSP Time [s]
c_1	0.05	0.0	0.0	0.4	0.15.
c_2	0.05	0.0	0.0	0.3	0.15
c_3	0.05	0.03	0.0	0.3	0.15
c_4	0.05	0.0	0.0	0.3	0.15
c_5	0.05	0.0	0.0	0.3	0.15
c_6	0.05	0.03	0.0	0.3	0.15
c_7	0.05	0.0	0.0	0.25	0.13
c_8	0.05	0.0	0.0	0.22	0.13
c_9	0.05	0.03	0.0	0.22	0.13
c_{10}	0.05	0.0	0.0	0.22	0.13
c_{11}	0.05	0.0	0.0	0.22	0.13
c_{12}	0.05	0.0	0.0	0.3	0.13
c_{13}	0.05	0.0	0.0	0.35	0.15
c_{14}	0.05	0.0	0.0	0.35	0.15
c_{15}	0.05	0.0	0.0	0.35	0.15

4 Conclusions

The real-time stabilization method, which optimizes the capture point trajectory with a CPT controller, was proved to be very effective when the robot has to implement dynamic walking commands, such as drastic dynamic walking, rotational dynamic walking, and fast dynamic walking commands. The drastic dynamic walking commands are characterized by short SSP times, short DSP times, large sagittal strides, and large lateral strides, and it was confirmed that the robot recovers its instability in a short period of time and implements all of the commands stably. The feasibility of these types of drastic dynamic walking commands is essential for the robot to work in environments where humans live with many dynamic moving obstacles. The stabilization method proposed here makes it possible for the robot to cope with dynamic environments. Rotational dynamic walking commands include rotational motions and it was confirmed that the system is more stable with the proposed stabilization method with a CPT controller than the conventional CPT controller-based walking algorithm. The fast dynamic walking commands have a very short SSP time and DSP time. Although it is very difficult to make these kinds of walking commands feasible, it was confirmed that the robot can walk fast and stably with the proposed stabilization method, whereas the conventional approach fails and the robot falls down.

The feasibility of implementing dynamic walking commands can be assured by optimizing the CP trajectory in every sampling time. The CP optimization generates the

optimal CP offset which was set by heuristically or by backward iteration method before. It can be seen that CP-based walking still holds when the optimization is applied, as shown in Fig. 14. This is because a CPT controller layer works to follow the optimal CP trajectory with an optimal control input generated near the support polygon. This assures that the robot recovers any instability as soon as the ZMP becomes unstable. In addition, the feasibility of the control input is assured without the projection method as the control input is generated inside the support polygon. We used 8ms for the control sampling time and the optimization was performed under 5ms even at worst cases and the average of the optimization time was 2.5ms. Thus, we think that the proposed method can also be used to bigger robots with control sampling time of 4˜5ms.

Our future work will be to reduce computation time for the proposed method by refining objective function, optimization variables, and search spaces as shorter control sampling time is advisable for the control performances.

Acknowledgments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2016R1C1B1006691).

References

1. Vukobratovic, M., Borovac, B.: Zero-moment point-thirty five years of its life. Int. J. Humanoid Robot. **1**(1), 157–173 (2004)
2. Kajita, S., Osamu, M., Muneharu, S.: Real-time 3D walking pattern generation for a biped robot with telescopic legs. In: Proc. IEEE Int. Conf. Robot. Autom., vol. 3, pp. 2299–2306 (2001)
3. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: A realtime pattern generator for biped walking. In: Proc. IEEE Int. Conf. Robot. Autom., vol. 1, pp. 31–37 (2002)
4. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of zero-moment point. In: Proc. IEEE Int. Conf. Robot. Autom., vol. 2, pp. 1620–1626 (2003)
5. Morisawa, M., Kajita, S., Kaneko, K., Harada, K., Kanehiro, F., Fujiwara, K., Hirukawa, H.: Pattern generation of biped walking constrained on parametric surface. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 2405–2410 (2005)
6. Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., Inoue, H.: Online generation of humanoid walking motion based on fast generation method of motion pattern that follows desired ZMP. In: Proc. IEEE Int. Conf. Intell. Robots Syst., pp. 2684–2689 (2002)
7. Erbatur, K., Kurt, O.: Natural ZMP trajectories for biped robot reference generation. IEEE Trans. Ind. Electron. **56**(3), 835–845 (2009)
8. Hong, Y.-D., Lee, B.-J., Kim, J.-H.: Command state-based modifiable walking pattern generation on an inclined plane in pitch and roll directions for humanoid robots. IEEE/ASME Trans. Mechatron. **16**(4), 783–789 (2011)
9. Hong, Y.-D., Kim, J.-H.: 3-D command state-based modifiable bipedal walking on uneven terrain. IEEE/ASME Trans. Mechatron. **18**(2), 657–663 (2013)
10. Hong, Y.-D., Lee, B.-J.: Experimental study on modifiable walking pattern generation for handling infeasible navigational commands. J. Elect. Eng. Technol. **10**(6), 2368–2375 (2015)
11. Nishiwaki, K., Kagami, S.: Simultaneous planning of CoM and ZMP based on the preview control method for online walking control. In: Proc.IEEE-RAS Int. Conf. Humanoid Robots, pp. 745–751 (2011)
12. Park, J.H., Kim, K.D.: Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. In: Proc. IEEE Int. Conf. Robotics Automation, Leuven, Belgium, pp. 3528–3533 (1998)
13. Sato, T., Sakaino, S., Ohnishi, K.: Real-time walking trajectory generation method with three-mass models at constant body height for three dimensional biped robots. IEEE Trans. Ind. Electron. **58**(2), 376–383 (2011)
14. Kajita, S., Morisawa, M., Miura, K., Nakaoka, S., Harada, K., Kaneko, K., Kanehiro, F., Yokoi, K.: Biped walking stabilization based on linear inverted pendulum tracking. In: Proc. on IEEE Int. Conf. on Intelligent Robots and Systems, pp. 4489–4496 (2010)
15. Campos-Macias, L. et al.: Stabilization method for dynamic gait in bipedal walking robots. In: IEEE Int. Conf. on Humanoid Robots (Humanoids), pp. 1276–1281 (2016)
16. Hu, K., Ott, C., Lee, D.: Learning and generalization of compensative zero-moment point trajectory for biped walking. IEEE Trans. Robot **32**(3), 717725 (2016)
17. Pratt, J., Carff, J., Drakunov, S., Goswami, A.: Capture point: A step toward humanoid push recovery. In: Proceedings of the international conference on humanoid robots. IEEE, pp. 200–207 (2006)
18. Hof, A.L.: The extrapolated center of mass concept suggests a simple control of balance in walking. Human Movement Sci. **27**, 112–125 (2008)
19. Englsberger, J., Ott, C., Roa, M.A., Albu-Schäffer, A., Hirzinger, G.: Bipedal walking control based on capture point dynamics. In: IEEE/RSJ int. conference on intelligent robots and systems, pp. 4420–4427 (2011)
20. Englsberger, J., Ott, C.: Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking. In: 2012 12th IEEE-RAS international conference on humanoid robots (Humanoids), pp. 183–189 (2012)
21. Morisawa, M., Kajita, S., Kanehiro, F., Kaneko, K., Miura, K., Yokoi, K.: Balance control based on capture point error compensation for biped walking on uneven terrain. In: IEEE-RAS Int. Conf. on Humanoid Robots, pp. 734–740 (2012)
22. Krause, M., Englsberger, J., Wieber, P.-B., Ott, C.: Stabilization of the capture point dynamics for bipedal walking based on model predictive control. IFAC Pro. **45**, 22 (2012)
23. Lanari, L., Hutchinson, S., Marchionni, L.: Boundedness issues in planning of locomotion trajectories for biped robots. In: Proc. of the IEEE/RSJ Int. Conf. on Humanoid Robots, pp. 951–958 (2014)
24. Lanari, L., Hutchinson, S.: Planning desired center of mass and zero moment point trajectories for bipedal locomotion. In: Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots, pp. 637–642 (2015)

In-Seok Kim received the B.S. and M.S. degrees in electrical and computer engineering from Ajou University, Suwon, Korea in 2016 and 2018, respectively. His current research interests include humanoid robotics, especially in bipedal walking pattern generation & control and path planning for robot navigation.

Young-Joong Han received the B.S. and M.S. degrees in electrical and computer engineering from Ajou University, Suwon, Korea in 2016 and 2018, respectively. His current research interests include humanoid robotics, especially in bipedal robot navigation and optimization based robot control.

Young-Dae Hong received the B.S., M.S., and Ph.D. degrees in electrical engineering from KAIST, Daejeon, Korea, in 2007, 2009, and 2013, respectively. Since 2014, he has been with the Department of Electrical and Computer Engineering, Ajou University, Korea, where he is currently an Assistant Professor. His current research interests include humanoid robotics, especially in bipedal walking pattern generation & control, footstep planning, and optimization based robot control.