Article

Capture Point-Based Controller Using Real-Time Zero Moment Point Manipulation for Stable Bipedal Walking in Human Environment

Young-Dae Hong

Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, Korea; ydhong@ajou.ac.kr

Received: 10 June 2019; Accepted: 2 August 2019; Published: 3 August 2019

Abstract: For collaboration of humans and bipedal robots in human environments, this paper proposes a stability control method for dynamically modifiable bipedal walking using a capture point (CP) tracking controller. A reasonable reference CP trajectory for the CP tracking control is generated using the real-time zero moment point (ZMP) manipulation without information on future footstep commands. This trajectory can be modified at any time during the single support phase according to a given footstep command. Accordingly, this makes it possible for the robot to walk stably with dynamically modifiable walking patterns, including sudden changes in navigational commands during the single support phase. A reference CP trajectory during the double support phase is also generated for continuity. The CP of the robot is controlled to track the reference trajectory using a ZMP-based CP tracking controller. The ZMP while walking is measured by the force-sensing resistor sensors mounted on the sole of each foot. A handling method for infeasible footstep commands is utilized so that the manipulated ZMP satisfies the allowable ZMP region for stability. The validity of the proposed method is verified through simulations and experiments.

Keywords: walking stability control; capture point (CP); zero moment point (ZMP) manipulation; modifiable walking pattern; 3-D linear inverted pendulum model (LIPM)

1. Introduction

Research on bipedal robots primarily focuses on achieving stable walking. Various approaches have been developed for stable walking pattern generation and control. One such approach generated a walking pattern by representing a bipedal robot as a simple three-dimensional (3-D) linear inverted pendulum model (LIPM) [1,2]. This method used the relationship between the center of mass (COM) of the robot and the zero moment point (ZMP) to generate a stable walking pattern. Based on the 3-D LIPM, walking pattern generation methods have been developed using specific forms of ZMP trajectories to improve the walking abilities of robots [3–5]. Some approaches have generated walking patterns using ZMP manipulation to achieve real-time modifiable bipedal walking [6–8].

Achieving stable bipedal walking based on a generated walking pattern requires a stability controller. An optimal controller was introduced to track a reference ZMP trajectory using preview data [9]. This controller was then extended to an auxiliary ZMP controller for walking on uneven terrain [10], as well as a model predictive controller that compensates for strong perturbation [11]. Subsequently, walking stability control methods based on a capture point (CP) were developed. The divergent component of COM motion is defined as the CP [12,13], meaning stable walking can be achieved by controlling the CP. A CP tracking controller based on the 3-D LIPM was proposed for walking stability control [14]. In this method, a ZMP-based CP tracking control method is utilized. A reference CP trajectory for the control is generated by heuristically setting the desired CP position at the end of a step. It was extended to a model containing vertical COM motion with angular momentum...
to improve control performance [15,16]. The reference CP trajectory generation by a backward calculation from the next three steps was also introduced. Furthermore, a walking controller based on divergent component of motion (DCM), which means CP in 3-D, was proposed for stable bipedal walking in 3-D environments [17].

Meanwhile, to achieve robust mobility in complex environments with dynamic obstacles, a bipedal robot should have the ability to modify its walking pattern dynamically in real-time. This ability enables a robot to respond to navigational commands that are changed in real-time from footstep planners [18,19]. For dynamically modifiable bipedal walking, the COM of the robot should be able to accelerate or decelerate freely during the single support phase without any restriction of its movement. A modifiable walking pattern generator (MWPG) was developed to facilitate independent changes in COM positions and velocities at any time using real-time ZMP manipulation [6–8]. Consequently, the real-time generation of modifiable walking patterns was realized. However, a walking stability controller is required to ensure the stability of dynamically modifiable bipedal walking.

This paper proposes a stability control method for dynamically modifiable bipedal walking using a CP tracking controller. For CP tracking controller, a reference CP trajectory is required. In the conventional CP tracking controllers, however, the reference CP trajectory is determined heuristically or determined by the iterative method to require information on command lists of several future footsteps, which makes it difficult to immediately respond to sudden changes in navigational commands during the single support phase. In this paper, these problems are solved by generating the reference CP trajectory using the real-time ZMP manipulation from the MWPG. Namely, the reference CP trajectory for stability control of dynamically modifiable walking can be generated in real-time without information on future footstep commands. This makes it possible for the robot to walk stably with the dynamically modifiable walking patterns, including sudden changes in navigational commands during the single support phase.

A command state is defined as a navigational command set of walking periods, step lengths, and walking rotations. The MWPG generates a modifiable walking pattern to achieve a given command state through ZMP manipulation. A reference CP trajectory for walking stability control is then generated using the real-time ZMP manipulation. This trajectory can be modified at any time during the single support phase according to the given command state, because it utilizes the real-time ZMP manipulation. Additionally, a reference CP trajectory in the double support phase is generated to facilitate a continuous CP trajectory. The CP of the robot is then controlled to track the generated reference CP trajectory. Occasionally, the manipulated ZMP may move outside of the stable ZMP region, which is within the foot region. In such cases, the walking pattern that satisfies the given command state causes unstable walking, meaning the command state is infeasible for the MWPG. Therefore, a handling method for the infeasible command states is utilized so that the manipulated ZMP satisfies the allowable ZMP region for stability. The proposed method is implemented with a small-sized bipedal robot, DARwIn-OP, and the validity of the proposed method is verified through simulations and experiments.

The remainder of this paper is organized as follows. Section 2 introduces the MWPG, and COM motion according to the command state is described, along with ZMP manipulation. Section 3 proposes the CP-based stability control method for dynamically modifiable walking. CP dynamics are briefly reviewed, and the generation of a reference CP trajectory in the single support phase using real-time ZMP manipulation is described. The generation of a reference CP trajectory in the double support phase and CP tracking control are also explained. The handling method for infeasible command states is described. Finally, the overall procedure of the proposed method is presented. Simulation and experimental results are provided and discussed in Section 4, and the paper is concluded in Section 5.
2. Modifiable Walking Pattern Generator

2.1. COM Motion of 3-D LIPM

A bipedal robot can be modeled as a 3-D LIPM, as shown in Figure 1. Its dynamic equation is written as follows:

\[ \ddot{\mathbf{x}} = \frac{\mathbf{x} - \mathbf{p}}{T_c^2} \]  

(1)

where \( \mathbf{x} = [x\ y]^T \) and \( \mathbf{p} = [p_x\ p_y]^T \) are the COM and the ZMP positions for sagittal and lateral motion, respectively. \( T_c = \sqrt{Z_c/g} \) is the time constant, and \( Z_c \) is the constant height of the 3-D LIPM. The sagittal and the lateral COM motions are derived from the dynamic equation of the 3-D LIPM, and the vertical COM motion is not considered to decouple the sagittal and the lateral COM motion equations. From the solution of Equation (1), the COM equation for the 3-D LIPM is obtained as follows [6–8]:

\[ \mathbf{x}_{\text{com}}(t) = \mathbf{A}(t)\mathbf{x}_{\text{com}0} + \mathbf{b}_p(t) \]

\[ \mathbf{y}_{\text{com}}(t) = \mathbf{A}(t)\mathbf{y}_{\text{com}0} + \mathbf{b}_q(t) \]  

(2)

with

\[ \mathbf{A}(t) = \begin{bmatrix} c(t) & s(t) \\ -s(t) & c(t) \end{bmatrix} \]

\[ \mathbf{b}_p(t) = \frac{1}{T_c} \begin{bmatrix} (s*p)(t) \\ (c*p)(t) \end{bmatrix} \]

\[ \mathbf{b}_q(t) = \frac{1}{T_c} \begin{bmatrix} (s*q)(t) \\ (c*q)(t) \end{bmatrix} \]

where \( \mathbf{x}_{\text{com}} = [x\ T_x]^T \) and \( \mathbf{y}_{\text{com}} = [y\ T_y]^T \) represent the walking state of the COM, which is defined by the COM position and the velocity in the sagittal and the lateral planes. \( \mathbf{x}_{\text{com}0} = [x_0\ T_x0]^T \) and \( \mathbf{y}_{\text{com}0} = [y_0\ T_y0]^T \) are the initial conditions of the walking state. \( c(t) \) and \( s(t) \) are abbreviations for \( \cos(h(t/T_c)) \) and \( \sin(h(t/T_c)) \), respectively. * denotes the convolution operator. \( p(t) \) and \( q(t) \) are the ZMP functions for sagittal and lateral COM motions, respectively, which are used for the real-time ZMP manipulation. If the ZMP is fixed on the center of the supporting foot, namely, \( \mathbf{p} = [0\ 0]^T \), in Equation (1), only the homogeneous solution part of Equation (2) is utilized, which results in predetermined and unmodifiable COM motion during the single support phase. Consequently, the COM of the robot is unable to accelerate or decelerate freely during the single support phase; accordingly, it is impossible to change walking period, step length, and walking rotation independently during the single support phase. To solve this problem, the ZMP functions, \( p(t) \) and \( q(t) \), are defined in the MWPG as the constant function for sagittal COM motion and the step function for lateral COM motion, respectively, as follows:

\[ p(t) = \begin{cases} P, & \text{if } 0 \leq t < T \\ 0, & \text{otherwise} \end{cases} \]

(3)

\[ q(t) = \begin{cases} Q, & \text{if } 0 \leq t < T_{sw} \\ -Q, & \text{if } T_{sw} \leq t < T \end{cases} \]

where \( P \) and \( Q \) denote the magnitudes of the ZMP functions, \( T \) denotes the remaining single support time, and \( T_{sw} \) denotes the switching time for the step function. Using these ZMP functions, real-time ZMP manipulation is facilitated during the single support phase. Namely, the COM motion during the single support phase is modifiable by using \( \mathbf{p} = [p(t)\ q(t)]^T \) in Equation (1) to satisfy the changes of walking period, step length, and walking rotation. The parameters for the ZMP manipulation, \( P, Q, T, \) and \( T_{sw} \), are calculated from Equation (2) [6–8]. Consequently, the COM position and the velocity can be independently changed at any time, which enables the real-time generation of dynamically modifiable walking patterns. However, the manipulation of the ZMP should be restricted to the region where the stability of the generated walking patterns is guaranteed. Therefore, the allowable ZMP region for stable walking patterns is defined by \( P_{\text{max}}/P_{\text{min}} \) and \( Q_{\text{max}}/Q_{\text{min}} \) within the foot region, as shown in Figure 2.
2.2. Desired Walking State from Command State

The following command state is defined as a navigational footstep command set:

\[ c = \begin{bmatrix} T_{ss}^{r} & T_{ss}^{l} & T_{ds}^{l} & T_{ds}^{r} & F_l & F_r & S_l & S_r & \theta_l & \theta_r \end{bmatrix}^T \]

where \( T_{ss}^{l/r} \), \( T_{ds}^{l/r} \), \( F_{l/r} \), \( S_{l/r} \), and \( \theta_{l/r} \) denote single support time, double support time, forward step length, side step length, and walking rotation angle for the left and the right legs, respectively. These parameters are defined with respect to the local coordinate frame attached on the support foot.

To generate a modifiable walking pattern, the desired walking state, represented by \( \mathbf{x}_{\text{ws}} \) and \( \mathbf{y}_{\text{ws}} \), is defined as the state of the COM at the end of each single support phase to satisfy the given command state. This walking state is uniquely obtained through the following equations, which can be derived from the walking configuration in the single and the double support phases [6–8]:

\[
\begin{align*}
\mathbf{x}_{\text{ws}}^l &= \begin{bmatrix} x_{\text{ws}l}^l \ y_{\text{ws}l}^l \end{bmatrix} = \left( A^*_l \ A^*_r - I_{4\times4} \right)^{-1} \left( A^*_l \mathbf{b}^*_l \ A^*_r \mathbf{b}^*_r \right) \\
\mathbf{x}_{\text{ws}}^r &= \begin{bmatrix} x_{\text{ws}r}^l \ y_{\text{ws}r}^l \end{bmatrix} = \left( A^*_l \ A^*_r - I_{4\times4} \right)^{-1} \left( A^*_l \mathbf{b}^*_l \ A^*_r \mathbf{b}^*_r \right)
\end{align*}
\]

with

\[
A^*_l = \begin{bmatrix} \cos \theta_{l/r} & -\sin \theta_{l/r} \\ \sin \theta_{l/r} & \cos \theta_{l/r} \end{bmatrix} \quad \text{and} \quad \mathbf{b}^*_l = \text{vec} \left( \begin{bmatrix} T_{ss}^{r/l} \ T_{ss}^{l/r} \ T_{ds}^{l/r} \ T_{ds}^{r/l} \ F_{l/r} \ S_{l/r} \ \cos \theta_{l/r} & -\sin \theta_{l/r} \\ \sin \theta_{l/r} & \cos \theta_{l/r} \end{bmatrix} \right)
\]

where \( \text{vec}(\cdot) \) represents the vectorization of the matrix by stacking the columns of the matrix into a single column vector. As shown in (4), information regarding the command state is involved in the matrices, \( A^*_l \) and \( \mathbf{b}^*_l \), which are used to calculate the desired walking state.
After obtaining the desired walking state, it is necessary to manipulate the ZMP for the transition from the current walking state to the desired walking state. To achieve the desired walking state, the required parameters for the ZMP functions in Equation (3), \( P, Q, T, \) and \( T_{sw} \), should be determined. These parameters can be obtained using Equation (2) [6–8]. Using these parameters, ZMP manipulation occurs whenever the command state changes, which enables the current walking state to reach the desired state by accelerating or decelerating the COM of the robot.

3. CP-Based Stability Control Method for Dynamically Modifiable Walking

3.1. CP Dynamics

The COM motion of the 3-D LIPM can be decomposed into convergent and divergent components. The CP refers to the divergent component of COM motion and is defined as follows [12,13]:

\[
\xi = x + T_c \dot{x}
\]

where \( \xi = [\xi_x, \xi_y]^T \) is the CP position and Equation (5) can be rewritten as follows:

\[
\dot{x} = \frac{\xi - x}{T_c}
\]

Equation (6) is stable because the pole of the transfer function is \(-1/T_c\), which indicates that the relationship between the CP and the COM is stable. Accordingly, the COM always follows the CP trajectory.

From the differentiation of Equation (5) using Equations (1) and (6), the following equation can be obtained:

\[
\ddot{\xi} = \dot{x} + T_c \ddot{x} = \frac{\xi - p}{T_c}.
\]

Equation (7) is unstable because the pole of the transfer function is \(1/T_c\), which indicates that the relationship between the ZMP and the CP is unstable. Namely, the CP diverges from the ZMP. Therefore, a ZMP-based CP tracking controller using an appropriate CP reference trajectory is required to control the divergent components of the COM motion for stable bipedal walking.

3.2. Reference CP Trajectory Generation in Single Support Phase Using Real-Time ZMP Manipulation

For the CP tracking controller, a reference CP trajectory is required. It can be obtained from the solution of Equation (7) as follows:

\[
\xi_{ref}(t) = e^{\frac{t}{T_c}} \xi + \left(1 - e^{\frac{t}{T_c}}\right)p.
\]

In the conventional CP tracking controllers, the initial CP and ZMP positions, \( \xi \) and \( p \), in Equation (8) for the reference CP trajectory generation, were heuristically determined by using some offset from the center of the supporting foot or determined by a backward iteration method using information on command lists of several future footsteps [14–17]. The conventional methods are able to cope with the change at each footstep of navigational command by recalculating the backward iteration method. However, it is difficult to immediately respond to sudden change of navigational command in the single support phase by using the backward iteration method, because the method requires a sequence of future footstep commands to generate the reference CP trajectory. In the proposed method, \( \xi \) and \( p \) are set by using the real-time ZMP manipulation from the MWPG without information on future footstep commands as follows:

\[
\xi = \xi_0 = x_0 + T_c \dot{x}_0
\]

\[
p = p_m = [p_{xm}, p_{ym}]^T
\]
with

\[ p_{xm} = \begin{cases} P, & \text{if } F_{l/r}^{\text{cur}} \neq F_{l/r}^{\text{pre}} \\ 0, & \text{otherwise} \end{cases} \]

\[ p_{ym} = \begin{cases} Q, & \text{if } S_{l/r}^{\text{cur}} \neq S_{l/r}^{\text{pre}} \\ -Q, & \text{else if } S_{l/r}^{\text{cur}} \neq S_{l/r}^{\text{pre}} \\ 0, & \text{otherwise} \end{cases} \]

where \( x_0 = [x_0, y_0]^T \) and \( \dot{x}_0 = [\dot{x}_0, \dot{y}_0]^T \) denote the initial COM position and the velocity of the nominal COM trajectory, respectively. \( F_{l/r}^{\text{cur}}/F_{l/r}^{\text{pre}} \) and \( S_{l/r}^{\text{cur}}/S_{l/r}^{\text{pre}} \) denote the current/previous forward and the side step lengths, respectively. Figure 3 illustrates the procedure for reference CP trajectory generation using ZMP manipulation. \( \xi \) is set to \( \xi_0 \), which is the initial CP position and is calculated using Equation (5) with \( x_0 \) and \( \dot{x}_0 \) from the MWPG. \( p \) is set to \( p_{\text{pre}} \), which is defined as a modifiable ZMP position and is determined by using the ZMP functions that are obtained to achieve the desired walking state derived from the given command state. This position can be changed at any time during the single support phase according to the command state. Specifically, it is modifiable within the allowable ZMP region (blue rectangle) and not fixed on the center of the supporting foot. Consequently, the reference CP trajectory can be generated in real-time through the ZMP manipulation, which facilitates real-time stability control for the dynamically modifiable walking pattern, including sudden changes in footstep commands during the single support phase.

![Reference capture point (CP) trajectory generation using ZMP manipulation.](image)

**Figure 3.** Reference capture point (CP) trajectory generation using ZMP manipulation.

### 3.3. Reference CP Trajectory Generation in Double Support Phase

For a continuous CP trajectory, the reference CP trajectory during the double support phase is generated through cubic spline interpolation as follows:

\[ \xi_{\text{ref}}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad (11) \]

with

\[ a_0 = \xi_{\text{ref}}(0), \quad a_1 = \dot{\xi}_{\text{ref}}(0) \]

\[ a_2 = \frac{3}{T_{ds}^{l/r}} \left[ \xi_{\text{ref}}(T_{ds}^{l/r}) - \dot{\xi}_{\text{ref}}(0) \right] \]

\[ - \frac{2}{T_{ds}^{l/r}} \left( \frac{\dot{\xi}_{\text{ref}}(0)}{T_{ds}^{l/r}} - \frac{\ddot{\xi}_{\text{ref}}(0)}{T_{ds}^{l/r}} \right) \]

\[ a_3 = - \frac{2}{T_{ds}^{l/r}} \left( \frac{\xi_{\text{ref}}(T_{ds}^{l/r}) - \dot{\xi}_{\text{ref}}(0)}{T_{ds}^{l/r}} \right) \]

\[ + \frac{\dot{\xi}_{\text{ref}}(T_{ds}^{l/r}) + \ddot{\xi}_{\text{ref}}(0)}{T_{ds}^{l/r}} \]

where \( \xi_{\text{ref}}(0) \) and \( \dot{\xi}_{\text{ref}}(0) \) are obtained from the nominal COM position and the velocity at the end of the single support phase, \( x_{T_{ss}}^{l/r} \) and \( \dot{x}_{T_{ss}}^{l/r} \). Because the nominal COM trajectory during the double support phase is generated to travel with constant velocity in the MWPG, \( \xi_{\text{ref}}(T_{ds}^{l/r})/\dot{\xi}_{\text{ref}}(T_{ds}^{l/r}) \) are obtained from the nominal COM position and the velocity at the end of the double support phase, \( x_{T_{ds}}^{l/r} \) and \( \dot{x}_{T_{ds}}^{l/r} \), as follows:
where

\[
\begin{align*}
\dot{x}_{r_{ds}} &= x_{r_{ds}} + T_c \dot{x}_{r_{ds}}, \\
\dot{y}_{r_{ds}} &= y_{r_{ds}} + T_c \dot{y}_{r_{ds}}
\end{align*}
\]

and

\[
\begin{align*}
\dot{x}_{p_{ds}} &= x_{p_{ds}} + T_c \dot{x}_{p_{ds}}, \\
\dot{y}_{p_{ds}} &= y_{p_{ds}} + T_c \dot{y}_{p_{ds}}
\end{align*}
\]

with

3.4. CP Tracking Control

In order to control the current CP to track the reference CP trajectory, the following control law is derived from Equation (8) [15]:

\[
p_d(t) = \frac{1}{1 - e^{\frac{-\Delta t}{\tau_c}}} \dot{x}_c(t) - \frac{\Delta t}{\tau_c} \ddot{x}_c,
\]

with

\[
\dot{x}_c(t) = \dot{x}_{ref}(t + \Delta T),
\]

where \( \dot{x}_c = \begin{bmatrix} \dot{x}_c \dot{y}_c \end{bmatrix}^T \) and \( \dot{x}_{ref} = \begin{bmatrix} \dot{x}_{ref} \dot{y}_{ref} \end{bmatrix}^T \) are the target and the current CPs, respectively. \( \Delta T \) is the sampling time for controlling the CP. \( p_d(t) \) is the required ZMP for the current CP to track the target CP. The force for the required ZMP can be obtained using Equation (1) as follows:

\[
F_d = m \frac{x - p_d}{\ell_c^2} = \frac{mg}{\ell_c}(x - p_d),
\]

where \( m \) is the mass of the COM. The required acceleration of the COM for the calculated force can be derived from the current force exerted on the COM, \( F_c \), as follows:

\[
\dot{x}_d = K_f(F_d - F_c),
\]

where \( K_f = \begin{bmatrix} k_x & 0; 0 & k_y \end{bmatrix} \) is the coefficient matrix. By inserting Equation (14) into Equation (15), the required acceleration of the COM for satisfying the desired ZMP can be rewritten as follows:

\[
\dot{x}_d = K_f \frac{mg}{\ell_c}(p_d - p_c),
\]

where \( p_c \) is the current ZMP. The target COM position and the velocity, \( x_t \) and \( \dot{x}_t \), are calculated by adding the integration of Equation (16) to the nominal COM trajectory from (2). Consequently, they enable the ZMP while walking to follow the desired ZMP for CP tracking control.

3.5. Handling Infeasible Command States and the Corresponding Desired Walking States

As mentioned in Section 2, the ZMP functions, \( p(t) \) and \( q(t) \), should stay within the allowable ZMP region where the stability of the generated walking patterns is guaranteed. If \( P \) and \( Q \) of the ZMP functions are larger/smaller than \( P_{max}/P_{min} \) and \( Q_{max}/Q_{min} \), then the desired walking state for a given command state cannot be achieved from the current walking state by using ZMP manipulation within the allowable region. Namely, the command state and the corresponding desired walking state are infeasible in the MWPG. Therefore, a handling method for the infeasible command states and corresponding desired walking states is required for the generation of a stable walking pattern. Command states provided by a footstep planner are entered as inputs to the MWPG at every sampling time. Some of these command states may be infeasible, since the footstep planner does not consider the feasibility of the planned footstep commands. \( f = [F_f F_s S_t \theta_t \theta_s] \) is defined as the placement information of a footstep. If an infeasible command state is given, \( f \) of the infeasible command state is modified to achieve a feasible command state through a binary search algorithm. A binary
search algorithm is appropriate for real-time walking pattern generation because it has a very low computational cost.

Algorithm 1 shows the pseudo code of the handling method, where \( f_{\text{pre}} \) and \( f_d \) refer to the footstep placement information of the previous and the desired command states, \( c_{\text{pre}} \) and \( c_d \), respectively. \( f(\cdot) \) denotes the function for calculating the desired walking state from the command state using (4). \( N \) is the total number of iterations. After initialization, the modified desired walking state, \( X_{\text{m}} \), is calculated using \( c_{\text{m}} \), where \( f_m \) is the mean value of \( f_0 \) and \( f_1 \). If \( X_{\text{m}} \) is feasible, then \( f_m \) is set to \( f_0 \) and the feasible command state and the desired walking state, \( c_f \) and \( X_{\text{ws}} \), are updated. Otherwise, \( f_m \) is set to \( f_1 \). These steps are repeated until the current iteration number, \( n \), reaches \( N \). If a feasible command state is not obtained before \( n \) reaches \( N \), the previous command state is used instead. The ZMP manipulation is not required for the same command state as the previous command state. Therefore, stable walking is guaranteed.

Algorithm 1. Handling infeasible command states and desired walking states.

/* initialization */
\[
\begin{align*}
&f_0 \leftarrow f_{\text{pre}}; \\
&f_1 \leftarrow f_d; \\
&c_f \leftarrow c_{\text{pre}}; \\
&X_{\text{ws}} \leftarrow f(c_f); \\
&n \leftarrow 0; \\
/* start binary search algorithm */
\end{align*}
\]

while \( n < N \) do
\[
\begin{align*}
&f_m \leftarrow (f_0 + f_1)/2; \\
&X_m \leftarrow f(c_m); \\
&\text{if } X_m \text{ is feasible then} \\
&\quad f_0 \leftarrow f_m; \\
&\quad c_f \leftarrow c_m; \\
&\quad X_{\text{ws}} \leftarrow X_m; \\
&\quad \text{break;} \\
&\text{else} \\
&\quad f_1 \leftarrow f_m; \\
&\end{align*}
\]
end if
\[
\begin{align*}
&n \leftarrow n + 1; \\
&\end{align*}
\]
end while

3.6. Overall Procedure

Figure 4 illustrates the overall procedure of the proposed method. The desired command state, \( c_d \), provided by a footstep planner is entered as an input to the proposed algorithm at every sampling time. The desired walking state, \( X_{\text{ws}} \), is obtained from \( c_d \) using Equation (4). The parameters of the ZMP functions, \( P, Q, T, \) and \( T_{sw} \), are calculated to achieve \( X_{\text{ws}} \) for the current walking state. The nominal COM trajectory, \( x/x \), is calculated using Equation (2) with the ZMP functions, \( p(l) \) and \( q(l) \), and the current walking state is updated. The feasibility of \( c_f \) and \( X_{\text{ws}} \) is determined by the allowable ZMP region, \( P_{\text{max}}/P_{\text{min}} \) and \( Q_{\text{max}}/Q_{\text{min}} \). If the given \( c_d \) is infeasible, it is substituted with a feasible command state, \( c_f \), obtained by Algorithm 1, and the corresponding desired walking state and the ZMP function parameters are recalculated. The reference CP trajectory in the single support phase, \( \xi_{\text{ref}} \), is generated using Equations (8)–(10). Note that \( \xi_{\text{ref}} \) in the double support phase is generated using Equations (11) and (12). The desired ZMP, \( p_d \), is calculated using Equation (13) with \( \xi_{\text{ref}} \) and the current CP, \( \xi_c \). \( p_d \) is then transferred to the ZMP controller, which generates the target COM position/velocity, \( x_t/x_t \), using Equation (16) with the current ZMP, \( p_c \). Subsequently, the trajectories for every leg joint of the robot are calculated by inverse kinematics. Note that the closed-form solution for inverse kinematics is derived.
by using algebra and geometry in the leg configuration of the bipedal robot. Therefore, the unique angles for every leg joint are obtained by the inverse kinematics.

4. Experimental Results and Discussion

The proposed method was implemented for a small-sized bipedal robot, DARwIn-OP. The height and the weight of the robot are 45.5 cm and 2.8 kg, respectively. DARwIn-OP has twenty degrees of freedom (DOFs) with Dynamixel MX-28 servo actuators (two DOFs for the head, six for the arms, and twelve for the legs). It uses an Intel Atom Z530 central processing unit (1.6 GHz) as the main controller, which calculates the proposed algorithm in real-time every 5 ms. Four force-sensing resistors (FSRs) are mounted on the sole of each foot to measure ground reaction forces (GRFs) and ZMPs while walking. The measurement range of the FSR is from 0.493 N to 65.535 N, and the unit of measurement is 0.001 N. The ZMP is calculated as follows:

\[
p = \frac{\sum_{i=1}^{4}(f_{ll}p_{ll} + f_{rr}p_{rr})}{\sum_{i=1}^{4}(f_{ll} + f_{rr})}
\]

(17)

where \(p_{ll}\) and \(p_{rr}\) (\(i = 1, 2, 3, \) and \(4\)) denote the \(2 \times 1\) position vectors of the FSRs of the left and the right feet, respectively. \(f_{ll}\) and \(f_{rr}\) denote the GRF measured by each FSR. Note that the proposed method is not limited to the application to DARwIn-OP.

4.1. Simulation Results

The simulations were performed using the DARwIn-OP simulation model by Webots, which is a 3-D dynamic robotics simulator [20]. Four FSRs were equipped on the sole of each foot in the DARwIn-OP simulation model to measure the ZMP in the simulations. The following footstep commands were realized for the simulation of modifiable bipedal walking, including a sudden adjustment of foot placement during the single support phase in which step lengths were independently changed while maintaining the same walking period for each footstep. The walking rotation angles were set to zero to verify the inherent performances of the proposed method without the influence of disturbances such as slip:

1. Initial command state, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 0.4 \ 0.4 \ 7.4 \ -7.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
2. After 2nd step, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 8.0 \ 8.0 \ 6.4 \ -7.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
   (2-1) After \(t = 0.2\) s at 3rd step, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 3.0 \ 3.0 \ 6.4 \ -7.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
3. After 3rd step, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 8.0 \ 8.0 \ 6.4 \ -10.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
4. After 4th step, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 4.0 \ 4.0 \ 7.4 \ -10.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
5. After 5th step, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 1.0 \ 1.0 \ 7.4 \ -10.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
6. After 6th step, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 2.0 \ 2.0 \ 7.4 \ -10.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
7. After 7th step, \(c = [0.4 \ 0.4 \ 0.4 \ 0.4 \ 1.0 \ 1.0 \ 7.4 \ -8.4 \ 0.0 \ 0.0 \ 0.0 \ 0.0]^{T}\)
where time, length, and angle units are given in seconds, centimeters, and degrees, respectively.

Figure 5 shows the snapshots of the first simulation result with the proposed CP-based walking stability controller, and Figure 6 illustrates the reference CP trajectory, \( \xi_{\text{ref}} \), generated through ZMP manipulation, as well as the measured CP and COM trajectories, \( \xi_c \) and \( x_c \), for the modifiable walking pattern in the first simulation. The initial CP position at each footstep (small green circle), \( \xi_0 \), was derived from the initial nominal COM position (small blue circle) and the velocity in the single support phase. The modifiable ZMP position (small red circle), \( p_m \), was derived from the ZMP functions for the given footstep command, which fell within the allowable ZMP region (thin rectangle) set as \( P_{\text{max}} = 3.0 \text{ cm}, P_{\text{min}} = -3.0 \text{ cm}, Q_{\text{max}} = 2.5 \text{ cm}, \) and \( Q_{\text{min}} = -1.5 \text{ cm} \).

\[
(8) \text{ After 8th step, } c = [0.4 \ 0.4 \ 0.4 \ -3.0 \ -3.0 \ 8.4 \ -8.4 \ 0.0 \ 0.0]^T \\
(9) \text{ After 9th step, } c = [0.4 \ 0.4 \ 0.4 \ -5.0 \ -5.0 \ 8.4 \ -7.4 \ 0.0 \ 0.0]^T \\
(10) \text{ After 10th step, } c = [0.4 \ 0.4 \ 0.4 \ 0.0 \ 0.0 \ 7.4 \ -7.4 \ 0.0 \ 0.0]^T 
\]
Figure 7 illustrates the variation in the parameters of the ZMP functions relative to the center of the supporting foot. The figure indicates that ZMP manipulation occurred whenever the footstep command changed. Therefore, the modifiable ZMP positions also changed through the ZMP manipulation. Specifically, the forward step length of the third footstep was modified from 8.0 cm to 3.0 cm at a time 0.2 s after the start of the single support phase of 0.4 s. The footstep command was modified in the single support phase, which resulted in ZMP manipulation. Figure 7 shows that the ZMP function for sagittal COM motion, p(t), was modified at t = 2.2 s. For the abruptly shortened step length, the parameter of the ZMP function, P, was changed, and the remaining single support time, T, was modified from 0.2 s to 0.09 s. From the modified ZMP function, the modifiable ZMP position was also changed to realize the modified footstep command. Figure 8 presents an enlarged view of the area around the second footstep in Figure 6. The modifiable ZMP position for the third footstep was changed to a more forward position compared to that for the forward step length before modification. Consequently, the reference CP trajectory was modified in real-time according to the changed modifiable ZMP position. The real CP and COM trajectories were also adjusted to track the modified reference CP trajectory, as shown in Figure 8.

![Figure 7. Variation in the parameters of the ZMP functions in the first simulation.](image)

![Figure 8. Variation in the modifiable ZMP position, p_m, as well as CP and COM trajectories, z_ref, z_cr, and x_s, when the command state for the third footstep changed during the single support phase.](image)

Figure 9 plots the reference and the measured CP trajectories in the x-axis and the y-axis. The real CP trajectory followed the reference trajectory with small error, as shown in Figure 10, which was facilitated by the CP tracking controller that enabled the current ZMP to follow the desired ZMP trajectory. In Figure 11, there is a peak in the desired ZMP trajectory for CP tracking control in the x-axis at t = 2.2 s. This peak was caused by the instant variation in the target CP resulting from the change in the footstep command. However, the following desired ZMP trajectory was generated normally, meaning the ZMP followed the desired trajectory continuously without being influenced by the peak. It can be seen that the ZMP was measured with a small variation from the foot center trajectory. Consequently, the real CP trajectory was able to successfully follow the reference CP trajectory for stable modifiable bipedal walking.
Figure 9. Reference and measured CP trajectories, $\xi_{ref}$ and $\xi_c$, in the first simulation.

Figure 10. Error trajectories between the reference and the measured CPs, $\xi_{ref}$ and $\xi_c$, in the first simulation.

Figure 11. Trajectories of desired ZMP for CP tracking control and measured ZMP, $p_d$ and $p_c$, in the first simulation.
To demonstrate the effectiveness of the infeasible command state handling method, an additional simulation was performed with the following footstep commands:

(1) Initial command state, \( c = [0.4 \ 0.4 \ 0.4 \ -9.0 \ -9.0 \ 7.4 \ -7.4 \ 0.0 \ 0.0]^T \)
(2) After 1st step, \( c = [0.4 \ 0.4 \ 0.4 \ -9.0 \ -9.0 \ 7.4 \ -10.4 \ 0.0 \ 0.0]^T \)
(3) After 2nd step, \( c = [0.4 \ 0.4 \ 0.4 \ 5.0 \ 5.0 \ 7.4 \ -10.4 \ 0.0 \ 0.0]^T \)
(4) After 3rd step, \( c = [0.4 \ 0.4 \ 0.4 \ 5.0 \ 5.0 \ 7.4 \ -7.4 \ 0.0 \ 0.0]^T \)
(5) After 5th step, \( c = [0.4 \ 0.4 \ 0.4 \ 0.0 \ 0.0 \ 7.4 \ -7.4 \ 0.0 \ 0.0]^T \)

DARwIn-OP has a relatively large foot for its leg length, which makes most footstep commands feasible because of its relatively large allowable ZMP region. Therefore, in this simulation, the allowable ZMP region was intentionally reduced to \( P_{\text{max}} = 0.3 \text{ cm}, P_{\text{min}} = -0.3 \text{ cm}, Q_{\text{max}} = 0.25 \text{ cm}, \) and \( Q_{\text{min}} = -0.15 \text{ cm} \) to simulate infeasible command state handling.

Figure 12 shows that the robot initially walked straight backward and then changed its walking rotation after the second footstep to move forward with a modified forward step length of 5.0 cm and side step length of −10.4 cm. At this moment, ZMP manipulation was required to achieve the desired walking state for the modified footstep command. However, the ZMP required by the manipulation fell outside of the reduced allowable ZMP region (small thick rectangle), meaning that the footstep command was infeasible. Therefore, the handling method modified the infeasible footstep command (pink cross) into a feasible footstep command for the third footstep (small black circle) with a forward step length of 2.9 cm and side step length of −9.9 cm. Accordingly, the parameters for the ZMP functions were modified to fall within the reduced allowable ZMP region, as shown in Figure 13. Consequently, a reference CP trajectory was generated through ZMP manipulation within the allowable region, and stable walking control was achieved. After the third footstep, the robot walked stably according to the given (feasible) footstep commands.

Figure 12. Reference CP trajectory, \( \xi_{\text{ref}} \), generated through ZMP manipulation, as well as the measured CP and COM trajectories, \( \xi_\text{m} \) and \( x_c \), for the modifiable walking pattern with an infeasible command state in the second simulation.

Figure 13. Variation in parameters for ZMP functions in the second simulation.
4.2. Experimental Results

Experiments were performed using a real bipedal robot, DARwIn-OP. The same footstep commands used in the first simulation were realized for the experiment on modifiable bipedal walking. A walking experiment that did not use the walking stability controller was performed first for the comparisons with the walking experiment that used the proposed walking stability controller. Figure 14 illustrates the snapshots of the experiment result without the walking stability controller. The robot walked unstably and almost fell as it stumbled at the fifth footstep, as shown in the Figure. In this experiment, CP tracking control was not performed, meaning the desired ZMP trajectory for CP tracking control, \( p_d \), was not obtained. Accordingly, the measured ZMP trajectory was compared to the nominal ZMP trajectory obtained from the ZMP functions of the MWPG relative to the global coordinate system, as shown in Figure 15. The figure indicates that the measured ZMP trajectory followed the nominal ZMP trajectory with significant variation, and it was sometimes saturated. This instability was caused by the dynamic COM motion from the MWPG needed to perform the given footstep commands. Consequently, it was confirmed that a walking stability controller was required to successfully realize stable modifiable bipedal walking, including a sudden adjustment of foot placement during the single support phase.

A second walking experiment was then performed using the proposed CP-based walking stability controller, as shown in Figure 16. Figure 17 illustrates the error trajectory between reference and measured CPs, \( \xi_{ref} \) and \( \xi_c \), in the experiment with the proposed CP-based walking stability controller. The real CP tracked the reference CP trajectory with a little error. The error between the reference and the measured CP trajectories was larger than in the simulation results. This occurred because of measurement noise, offsets, and unmodeled errors in the real robot, including joint compliance. Accordingly, the desired ZMP trajectory for CP tracking control was considerably perturbed to allow the real CP to track the reference trajectory, as shown in Figure 18. However, the ZMP successfully followed the desired trajectory by using the controller with a little variation from the foot center trajectory. Therefore, the real CP trajectory was able to follow the reference CP trajectory. Consequently, the robot was controlled to walk stably according to the dynamically modifiable walking patterns provided by the footstep commands.

Figure 14. Snapshots of the experiment result without the walking stability controller (left to right, top to bottom).
Figure 15. Trajectories of nominal ZMP obtained from the ZMP functions and measured ZMP, $p_c$, in the experiment without the walking stability controller.

Figure 16. Snapshots of the experiment result with the proposed CP-based walking stability controller (left to right, top to bottom).
Figure 16. Snapshots of the experiment result with the proposed CP-based walking stability controller (left to right, top to bottom).

Figure 17. Error trajectory between reference and measured CPs, $\xi_{\text{ref}}$ and $\xi_c$, in the experiment with the proposed CP-based walking stability controller.

Figure 18. Trajectories of the desired ZMP for CP tracking control and measured ZMP, $p_d$ and $p_c$, in the experiment with the proposed CP-based walking stability controller.

5. Conclusions

This paper proposed a CP-based walking stability controller to allow a bipedal robot to walk stably with a dynamically modifiable walking pattern. Infeasible footstep commands input to the MWPG were modified into feasible footstep commands through the handling method. Then, a reference CP trajectory for the single support phase was generated in real-time through ZMP manipulation without information on future footstep commands. This trajectory could be changed at any time according to the given footstep commands, which facilitated CP-based walking stability control to respond to sudden changes in navigational commands during the single support phase. The reference CP trajectory during the double support phase was also generated for continuity, and the CP of the robot was controlled to track the reference trajectory. The effectiveness of the proposed method was demonstrated through simulations and experiments.
Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2019R1C1C1002049).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Kajita, S.; Kanehiro, F.; Kaneko, K.; Fujiwara, K.; Yokoi, K.; Hirukawa, H. A realtime pattern generator for biped walking. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 1, pp. 31–37.
2. Sugihara, T.; Nakamura, Y.; Inoue, H. Realtime humanoid motion generation through ZMP manipulation based on inverted pendulum control. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; Volume 2, pp. 1404–1409.
3. Nishiwaki, K.; Kagami, S.; Kuniyoshi, Y.; Inaba, M.; Inoue, H. Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired ZMP. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; pp. 2684–2689.
4. Harada, K.; Kajita, S.; Kaneko, K.; Hirukawa, H. An analytical method for real-time gait planning for humanoid robots. Int. J. Hum. Robot. 2006, 2, 1–19. [CrossRef]
5. Erbatur, K.; Kurt, O. Natural ZMP trajectories for biped robot reference generation. IEEE Trans. Ind. Electron. 2009, 56, 835–845. [CrossRef]
6. Lee, B.-J.; Stonier, D.; Kim, Y.-D.; Yoo, J.-K.; Kim, J.-H. Modifiable walking pattern of a humanoid robot by using allowable ZMP variation. IEEE Trans. Robot. 2008, 24, 917–925.
7. 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. 2011, 16, 783–789. [CrossRef]
8. Hong, Y.-D.; Kim, J.-H. 3-D command state-based modifiable bipedal walking on uneven terrain. IEEE/ASME Trans. Mechatron. 2013, 18, 657–663. [CrossRef]
9. 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 Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; pp. 14–19.
10. Kajita, S.; Morisawa, M.; Harada, K.; Kaneko, K.; Kanehiro, F.; Fujiwara, K.; Hirukawa, H. Biped walking pattern generation allowing auxiliary ZMP control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 2993–2999.
11. Wieber, P.-B. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In Proceedings of the 2006 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, 4–6 December 2006; pp. 137–142.
12. Pratt, J.; Carff, J.; Drakunov, S.; Goswami, A. Capture point: A step toward humanoid push recovery. In Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, 4–6 December 2006; pp. 200–207.
13. Takenaka, T.; Matsumoto, T.; Yoshiike, T. Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 1084–1091.
14. Englsberger, J.; Ott, C.; Roa, M.A.; Albu-Schäffer, A.; Hirzinger, G. Bipedal walking control based on capture point dynamics. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4420–4427.
15. Englsberger, J.; Ott, C. Integration of vertical COM motion and angular momentum in an extended capture point tracking controller for bipedal walking. In Proceedings of the 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), Osaka, Japan, 29 November–1 December 2012; pp. 183–189.
16. Seyde, T.; Shrivastava, A.; Englsberger, J.; Bertrand, S.; Pratt, J.; Griffin, R.J. Inclusion of angular momentum during planning for capture point based walking. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1791–1798.
17. Englsberger, J.; Ott, C.; Albu-Schäffer, A. Three-dimensional bipedal walking control based on divergent component of motion. IEEE Trans. Robot. 2015, 31, 355–368. [CrossRef]
18. Chestnutt, J.; Nishiwaki, K.; Kuffner, J.; Kagami, S. An adaptive action model for legged navigation planning. In Proceedings of the 2007 7th IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, PA, USA, 29 November–1 December 2007; pp. 196–202.

19. Karkowski, P.; Oßwald, S.; Bennewitz, M. Real-time footstep planning in 3D environments. In Proceedings of the IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 15–17 November 2016; pp. 69–74.

20. Michel, O. Cyberbotics Ltd. WebotsTM: Professional mobile robot simulation. Int. J. Adv. Robot. Syst. 2004, 1, 39–42. [CrossRef]