

Dynamic Simulation of Modifiable Walking Pattern Generation to Handle Infeasible Navigational Commands for Humanoid Robots

Young-Dae Hong*, Ki-Baek Lee** and Bumjoo Lee[†]

Abstract – The modifiable walking pattern generation (MWPG) algorithm can handle dynamic walking commands by changing the walking period, step length, and direction independently. When an infeasible command is given, the algorithm changes the command to a feasible one. After the feasibility of the navigational command is checked, it is translated into the desired center of mass (CM) state. To achieve the desired CM state, a reference CM trajectory is generated using predefined zero moment point (ZMP) functions. Based on the proposed algorithm, various complex walking patterns were generated, including backward and sideways walking. The effectiveness of the patterns was verified in dynamic simulations using the Webots simulator.

Keywords: Humanoid robot, Bipedal robot, Walking pattern, Gait, Locomotion

1. Introduction

Most control algorithms for bipedal walking utilize a dynamic model of the robot, which is expressed by a differential algebraic equation (DAE). Because the DAE is generally difficult to solve in a closed form, many approaches have taken to simplifying the model and/or adopting an iterative search algorithm to generate a walking pattern. To implement a walking pattern in real time, reduced dynamic models such as the single linear inverted pendulum are commonly utilized. Because the inverted pendulum model effectively reflects the dominant dynamics of the walking motion, many successful results have been reported [1-4]. Other approaches reduce the complexity of the equation of motion by assuming that the zero moment point (ZMP) trajectory has a specific form [5-8]. Several methods do not use explicit walking trajectories. These methods strongly depend on a sensory feedback algorithm with several intuitive rules [9-11]. Other approaches are based on batch-type processing to predesign the walking trajectory and/or control policy. These approaches commonly utilize iterative computation to design either entire trajectories or piecewise trajectories over some time interval. As the computing power grows, these approaches become more tractable, and many successful results have been produced [12-15].

To increase the usability of humanoid robots, they should be able to walk in complex environments such as spaces with moving obstacles and/or narrow foothold areas. In other words, the walking algorithm should be able to

handle complex navigational commands. In previous works, to walk in complex environments, there were walking algorithms to generate the walking pattern by using a preview controller which is a ZMP tracking servo controller [14] and by using dynamically stable mixture and connection of pre-designed motions [16]. This was also achieved through the realization of the modifiable walking pattern generation (MWPG) algorithm [17], which was then improved to cover infeasible commands [18]. If the robot is assumed to be a linear inverted pendulum, the equations of motion between the ZMP and center of mass (CM) state are easily derived and analyzed. To generate periodic walking patterns and the transitional pattern between two periodic patterns in real time, analytical solutions of the equation of motion for the simplified dynamics are utilized. In the proposed algorithm presented in this paper, the ZMP variation scheme is adopted to incorporate the ability to modify the walking pattern without any extra steps for adjusting the CM motion. This makes it possible to determine whether a given navigational command is feasible or infeasible. When an infeasible command is given, the proposed algorithm changes the command to a feasible one by a binary search algorithm. After the feasibility of the navigational command is checked, it is translated into the desired CM state. To achieve the desired CM state, a reference CM trajectory is generated using predefined ZMP functions. Based on the proposed algorithm, various complex walking patterns were generated, including backward and sideways walking. The effectiveness of the patterns was verified in dynamic simulations using the Webots simulator. Note that the proposed algorithm is focused on feedforward pattern generation. For stable robot walking, the feedback controllers introduced in [19-22] need to be adopted.

This paper is organized as follows. Section 2 briefly reviews the MWPG algorithm. The walking motion was

* Corresponding Author: Dept. of Electrical Engineering, Myongji University, Korea. (bjlee@mju.ac.kr).

** Dept. of Electrical and Computer Engineering, Ajou University, Korea. (ydhong@ajou.ac.kr).

† Dept. of Electrical Engineering, Kwangwoon University, Korea. (kblee@kw.ac.kr).

Received: February 3, 2015; Accepted: December 10, 2015

analyzed in terms of the periodic motion and transition between different periodic motions. The ZMP variation method and infeasible handling scheme are also described. Section 3 presents the generation and examination of various test walking patterns. Section 4 presents the conclusions.

2. Review of Modifiable Walking Pattern

2.1 CM motion for modifiable walking pattern

Because the dynamic nature of a humanoid robot is generally formulated as a DAE, it is difficult to solve the dynamic equation analytically. Therefore, a simplified model is often utilized to reduce the system dynamics. The subsequent equation of motion can be simplified enough to be derived analytically. Fig. 1 shows a linear inverted pendulum referred to as the 3D linear inverted pendulum model (3D-LIPM), which was proposed in [1].

$$\begin{aligned} \mathbf{x} &= \mathbf{A}(t)\mathbf{x}_0 + \mathbf{b}_p(t) \\ \mathbf{y} &= \mathbf{A}(t)\mathbf{y}_0 + \mathbf{b}_q(t) \end{aligned} \quad (1)$$

where

$$\begin{aligned} \mathbf{x} &\triangleq \begin{bmatrix} x \\ T_c \dot{x} \end{bmatrix}, \quad \mathbf{y} \triangleq \begin{bmatrix} y \\ T_c \dot{y} \end{bmatrix} \\ \mathbf{A}(t) &\triangleq \begin{bmatrix} c(t) & s(t) \\ s(t) & c(t) \end{bmatrix}, \quad \mathbf{b}_p(t) = -\frac{1}{T_c} \begin{bmatrix} (s * p)(t) \\ (c * p)(t) \end{bmatrix}, \\ \mathbf{b}_q(t) &\triangleq -\frac{1}{T_c} \begin{bmatrix} (s * q)(t) \\ (c * q)(t) \end{bmatrix}, \end{aligned}$$

$c(t)$ and $s(t)$ are abbreviations of $\cosh(t/T_c)$ and $\sinh(t/T_c)$, respectively, with a time constant $T_c = \sqrt{Z_c/g}$ and constant height Z_c . $p(t)$ and $q(t)$ represent the ZMP function on the sagittal and lateral planes, respectively. Lastly, $*$ is the convolution operator. Despite its conciseness, Eq. (1) characterizes dominant dynamics and separates sagittal and lateral motions.

Note that the particular solution part relaxes the motion predetermined by the homogeneous solution part. By

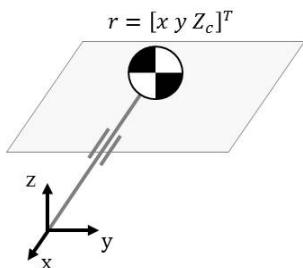


Fig. 1. Linear inverted pendulum model.

allowing variation in the ZMP for the bounded region, the position and velocity of the CM can be changed independently throughout the single-support phase.

For the sake of convenience, state space variables of the equation of motion that represents the walking pattern at a given time are defined as the *walking state* (*WS*). In the linear inverted pendulum model, the position and velocity of the CM become the *WS* as follows:

Definition 1 The *WS* for a linear inverted pendulum represents the CM state as follows:

$$\begin{aligned} \mathbf{x} &\triangleq [x \ T_c \dot{x}]^T \text{ for the sagittal motion,} \\ \mathbf{y} &\triangleq [y \ T_c \dot{y}]^T \text{ for the lateral motion.} \end{aligned}$$

2.2 Derivation of desired WS

The minimal set of parameters that includes the single- and double-support times, forward and side step lengths, and walking directions is defined as a *command state* (*CS*).

Definition 2 The *CS* is a minimal set that represents navigational commands as follows:

$$\mathbf{c} \triangleq [T_{sl} \ T_{sr} \ T_{dl} \ T_{dr} \ F_l \ F_r \ S_l \ S_r \ \theta_l \ \theta_r]^T.$$

where T_{sl} , T_{dl} , F_l , S_l , and θ_l represent the single-support time, double-support time, forward step length, side step length, and walking direction, respectively, for the left side. Similarly, T_{sr} , T_{dr} , F_r , S_r , and θ_r are for the right side. Note that the side step lengths in the lateral plane are for sideways walking.

A walking pattern can be modified in two ways: generating another periodic walking pattern that satisfies a given new *CS* or a transitional motion of the *WS* from the current state to the desired state on the new trajectory.

The *desired WS* is uniquely determined from a given *CS* and the *current WS* as shown in Fig. 2 which describes a periodic walking pattern according to the *CS*. In the figure, $\mathbf{Z} \triangleq [\mathbf{x} \ \mathbf{y}]$, and the subscripts $l_{(f)}$ and $r_{(f)}$ mean the left initial (final) and right initial (final) states, respectively, of the single support. Lastly, θ implies the walking direction.

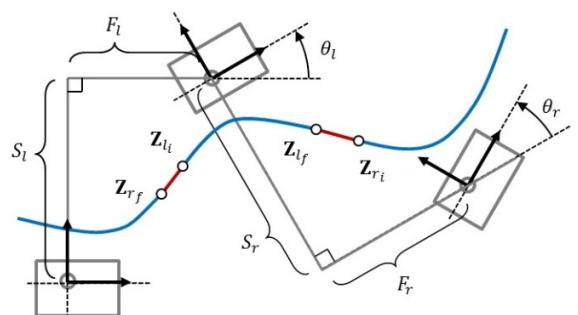


Fig. 2. Footprint of steady walking.

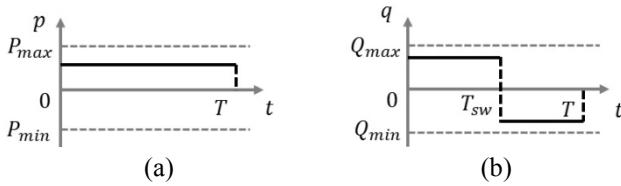


Fig. 3. ZMP functions: (a) sagittal function, (b) lateral function.

$$\begin{aligned}\zeta_{lf} &= (\mathbf{A}_r^* \mathbf{A}_l^* - \mathbf{I}_{4 \times 4})^{-1} (\mathbf{A}_r^* \mathbf{b}_l^* + \mathbf{b}_r^*) \\ \zeta_{rf} &= (\mathbf{A}_l^* \mathbf{A}_r^* - \mathbf{I}_{4 \times 4})^{-1} (\mathbf{A}_l^* \mathbf{b}_r^* + \mathbf{b}_l^*)\end{aligned}\quad (2)$$

Eq. (2) expresses the mapping relationship between the CS and *desired WS*. Whenever the CS is changed, it is translated into the *desired WS* form. Subsequently, ζ_{lf} or ζ_{rf} becomes the *desired WS* for the left or right support phase, respectively. Note that the information of the CS is involved in matrix \mathbf{A}^* and \mathbf{b}^* .

2.3 ZMP variation

After the *desired WS* is computed, the ZMP trajectory is adjusted to transfer the *current WS* to the desired one. Via the ZMP variation with closed-form functions, walking patterns are modified as intended by the CS. As introduced in the previous work, a constant function (Fig. 3 (a)) and step function (Fig. 3 (b)) are utilized for the sagittal and lateral motions, respectively. Detailed analysis of these two ZMP functions is covered in [16].

2.4 Binary search for infeasible CS processing

When the CS is infeasible, the CS and subsequent

Algorithm 1 Feasible CS and corresponding desired WS

```
/* initialize variables */
c0 ← cp; // set the previous CS as the starting point
c1 ← cd; // set the desired CS as the ending point
cf ← cp; // save the feasible CS
ζf ← f(cf); // feasible desired WS
n ← 0;
/* start binary search algorithm */
while n < N do
    cm ← (c0 + c1) / 2; // compute the middle point
    ζm ← f(cm); // computed the desired WS
    if ζm is feasible then
        c0 ← cm; // change the starting point
        /* update CS and desired WS */
        cf ← cm;
        ζf ← ζm;
    Else
        c1 ← cm; // change the ending point
    end if
    n ← n + 1;
end while
```

desired WS are simultaneously modified with the *binary search algorithm* given in Algorithm 1, where subscripts p and d represent the *previous* and *desired* states, respectively, and m and f represent the *modified* and *feasible* states, respectively. Lastly, $f(\cdot)$ means the function given a *desired WS* from a CS by Eq. (2). After initialization, CS is updated as a mean value between the previous and desired CS. Because the current WS is on the walking pattern generated by the previous CS, the *modified CS* is always feasible. Therefore, when the modified *desired WS* is feasible, then the modified *desired WS* moves to the first intended infeasible one. Otherwise, it moves to the previous one until the iteration reaches the maximum. This *binary search algorithm* is sufficiently fast to execute in real time because of its simplicity.

3. Computer Simulation

The proposed algorithm was simulated with a model of the small-sized humanoid platform DARwIn-OP to verify its effectiveness, as shown in Fig. 4(a). The height and weight of this platform are 45.5 cm and 2.8 kg, respectively. The walking simulation of DARwIn-OP was carried out using Webots, which is a 3D robotics simulation software that enables users to perform physical and dynamical simulations [23]. The model is shown in Fig. 4(b). Table 1 summarizes the CM height (Z_c), foot size (F_x , F_y), distance between foot centers (d_y), and ZMP allowable region ($P_{max/min}$, $Q_{max/min}$). Note that all length units are given in centimeters.

3.1 Backward walking

Backward walking patterns were realized from the following CS list (see the Definition 1):

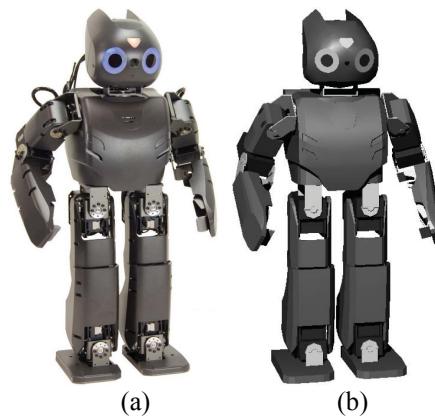


Fig. 4. (a) DARwIn-OP. (b) Simulation model.

Table 1. Simulation parameters.

Z_c	F_x	F_y	d_y	P_{max}	P_{min}	Q_{max}	Q_{min}
20.35	10.4	6.6	7.4	1.25	-1.25	0.75	-0.75

Table 2. Sagittal step lengths during backward walking.

Step	Current x-axis position	Past x-axis position	Sagittal step length
1	7.0	0.0	7.0
2	14.0	7.0	7.0
3	21.0	14.0	7.0
4	18.4	21.0	-2.6
5	10.4	18.4	-8.0
6	2.4	10.4	-8.0
7	-5.6	2.4	-8.0
8	-13.6	-5.6	-8.0
9	-13.6	-13.6	0.0

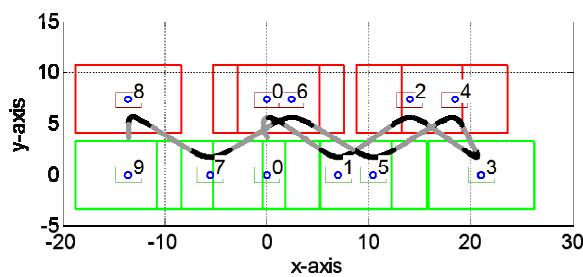


Fig. 5. Walking pattern (backward walking).

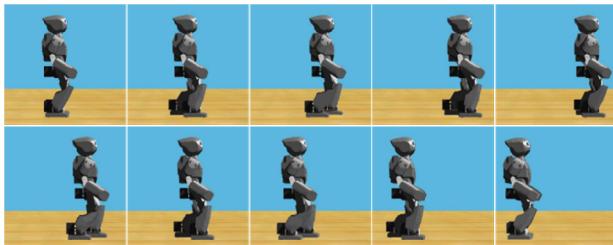


Fig. 6. Snapshots of backward walking simulation results (left to right, top to bottom).

- (1) Initial CS
 $\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 7.0 \ 7.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$
- (2) After the third step
 $\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ -8.0 \ -8.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$
- (3) After the seventh step
 $\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$

The time and length are given in seconds and centimeters, respectively, and the angle is given in degrees. Fig. 5 shows the backward walking pattern generated by the proposed method. The thick and thin curves represent the CM trajectories in the single- and double-support phases, respectively. The thick and thin rectangles represent the foot polygon and ZMP region, respectively. The circle and number mean the center of foot and step number, respectively. Table 2 lists the current/past x-axis positions of the foot centers and the sagittal step lengths at each footstep during the backward walking. According to the commanded CS list, the sagittal step length was changed to -8.0 cm after the third step for the backward walking. However, the sagittal step length was -2.6

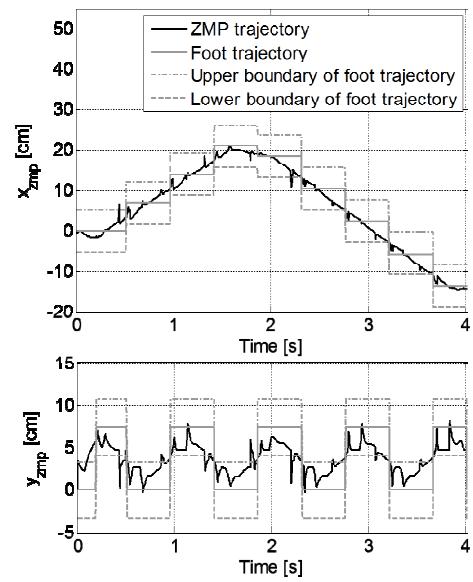


Fig. 7. Measured ZMP trajectories during backward walking simulation.

(=18.4 - 21.0) cm at the fourth step, and the commanded step length of -8.0 (=10.4 - 18.4) cm was achieved at the fifth step. This was because the infeasible CS commanded after the third step was replaced with a feasible CS by the proposed binary search algorithm. Fig. 6 presents snapshots of the backward walking simulation results. The humanoid robot successfully walked according to the feasible CSs with the proposed method. The ZMP trajectories during the walking simulation were measured and are shown in Fig. 7. The ZMP trajectories along the x- and y-axes followed the foot trajectories with a small variation. The walking of the humanoid robot consists of single and double support phases. For the stable dynamic walking, in the single support phase, the ZMP trajectory should be in the foot boundary of the support leg. However, in the double support phase, if the ZMP trajectory is in the area bounded by the two feet that are touching the ground, the robot is stable. In the figure, when the ZMP trajectories in the y-axis went outside of the foot boundaries, the robot was in the double support phase and thus the robot was stable.

3.2 Sideways walking

To show the effectiveness of the binary search algorithm, sideways walking patterns were also examined from the following CS list:

- (1) Initial CS
 $\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 6.0 \ 6.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$
- (2) After the second step
 $\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 2.0 \ 2.0 \ 7.4 \ -12.4 \ 0.0^\circ \ 0.0^\circ]^T$
- (3) After the seventh step
 $\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 2.0 \ 2.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$

(4) After the eighth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

Fig. 8 shows the footprints and CM trajectories of the sideways walking pattern generated from the CS list. Table 3 lists the current/past y-axis positions of the foot centers and the lateral step lengths at each footstep during the sideways walking. The lateral step length of the right leg was changed to -12.4 cm after the second step. However, it was not satisfied at the third step (-11.6 ($= -4.2 - 7.4$) cm). The commanded step length of -12.4 ($= -9.2 - 3.2$) cm was achieved at the fifth step. Figs. 9 and 10 present snapshots of the sideways walking simulation results and the measured ZMP trajectories. Similar to the backward walking simulation, the humanoid robot walked according

Table 3. Lateral step lengths during sideways walking.

Step	Current y-axis position	Past y-axis position	Lateral step length
1	0.0	7.4	-7.4
2	7.4	0.0	7.4
3	-4.2	7.4	-11.6
4	3.2	-4.2	7.4
5	-9.2	3.2	-12.4
6	-1.8	-9.2	7.4
7	-14.2	-1.8	-12.4
8	-6.8	-14.2	7.4
9	-14.2	-6.8	-7.4

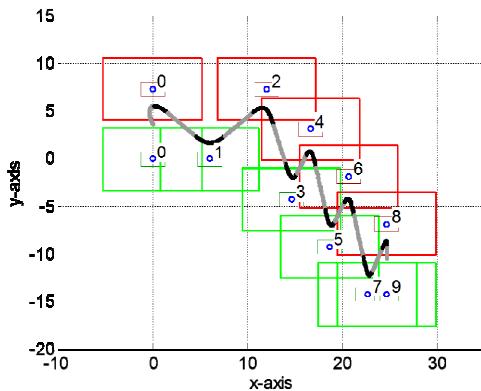


Fig. 8. Walking pattern (sideways walking).

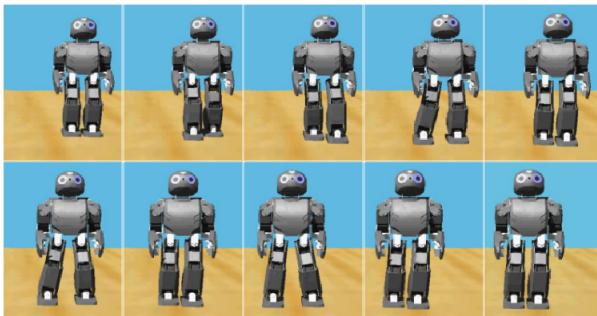


Fig. 9. Snapshots of sideways walking simulation results (left to right, top to bottom).

to the feasible CSs, and the ZMP trajectories followed the foot trajectories successfully.

3.3 Complex walking

As previously stated, the MWPG algorithm can generate a walking pattern even if the navigational command is complex. To show this ability, two complex walking simulations were carried out. In the first walking pattern, the CS included forward, backward, sideways, and turning motions:

(1) Initial CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 8.0 \ 8.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(2) After the second step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 4.0 \ 4.0 \ 7.4 \ -7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(3) After the fourth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 4.0 \ 4.0 \ 7.4 \ -9.4 \ 0.0^\circ \ -20.0^\circ]^T$$

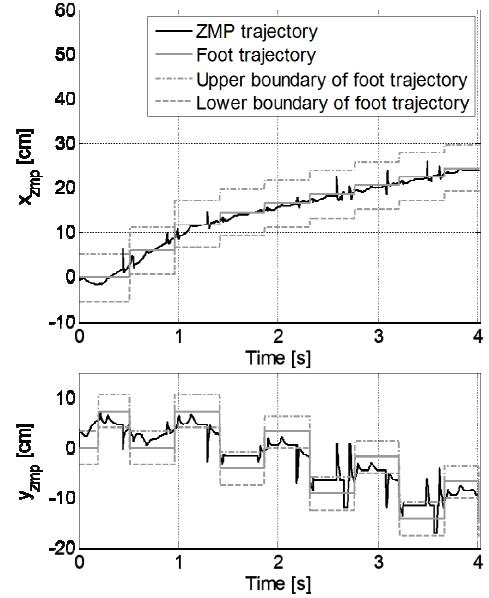


Fig. 10. Measured ZMP trajectories during sideways walking simulation.

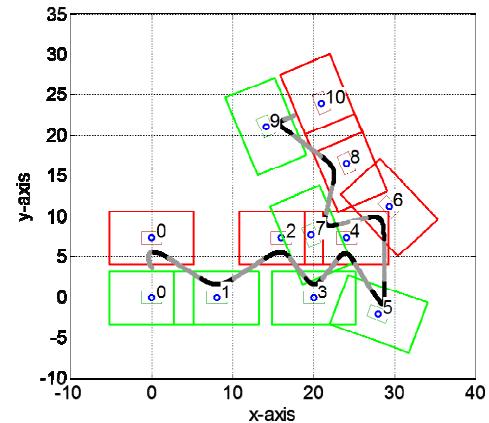


Fig. 11. Walking pattern (complex walking: case 1).

(4) After the fifth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ -4.0 \ -4.0 \ 1.3 \ 4 \ -7.4 \ -30.0 \ 0.0]^T$$

(5) After the sixth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ -4.0 \ -4.0 \ 0.7 \ 4 \ -9.4 \ 0.0 \ -20.0]^T$$

(6) After the seventh step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ -8.0 \ -8.0 \ 0.7 \ 4 \ -7.4 \ 0.0 \ 0.0]^T$$

(7) After the ninth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 0.7 \ 4 \ -7.4 \ 0.0 \ 0.0]^T$$

Fig. 11 shows the first complex walking pattern including forward, backward, sideways walking, and turning motions. After the second step, the sagittal step length was shortened. The walking direction was changed to the right with a different lateral step length after the fourth step. After the fifth step, backward walking was started by changing the sagittal and lateral step lengths and the walking direction.

In the second walking pattern, asymmetric forward walking motions were realized from the following CS list:

(1) Initial CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 6.0 \ 6.0 \ 0.7 \ 4 \ -7.4 \ 0.0 \ 0.0]^T$$

(2) After the third step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 0.0 \ 7.4 \ -7.4 \ 0.0 \ 0.0]^T$$

(3) After the fourth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 9.0 \ 9.0 \ 0.7 \ 4 \ -7.4 \ 0.0 \ 0.0]^T$$

(4) After the fifth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 0.0 \ 7.4 \ -7.4 \ 0.0 \ 0.0]^T$$

(5) After the sixth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 9.0 \ 9.0 \ 0.7 \ 4 \ -7.4 \ 0.0 \ 0.0]^T$$

(6) After the seventh step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 0.0 \ 7.4 \ -7.4 \ 0.0 \ 0.0]^T$$

Fig. 12 shows the second complex walking pattern. In this case, after the third step, the sagittal step length of the left leg was changed to 0.0 cm, and the sagittal step length of the right leg was changed to 9.0 cm. Figs. 13 and 14 present the measured ZMP trajectories during the complex walking simulations. The ZMP trajectories followed the foot trajectories.

3.4 Variation of step length according to allowable ZMP region

To analyze the effect when the allowable ZMP region is

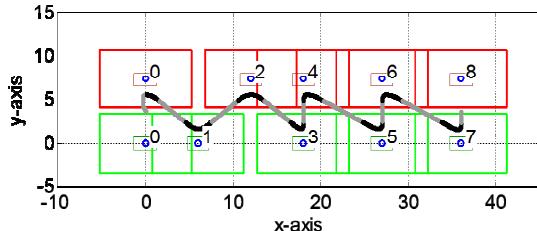


Fig. 12. Walking pattern (complex walking: case 2)

changed, the walking simulations were carried out varying the size of the allowable ZMP region in the sagittal plane, $P (P = P_{max} - P_{min})$. In the simulations, P was set to 4.5, 3.5, 2.5, 1.5, and 1.0 cm. The commanded CS list was changed dramatically as follows:

(1) Previous CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 1.0 \ 1.0 \ 0.7 \ 4 \ -7.4 \ 0.0 \ 0.0]^T$$

(2) New CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 1.5 \ 0.1 \ 5.0 \ 0.7 \ 4 \ -7.4 \ 0.0 \ 0.0]^T$$

Table 4 shows the simulation results. According to the commanded CS list, the sagittal step length was changed to

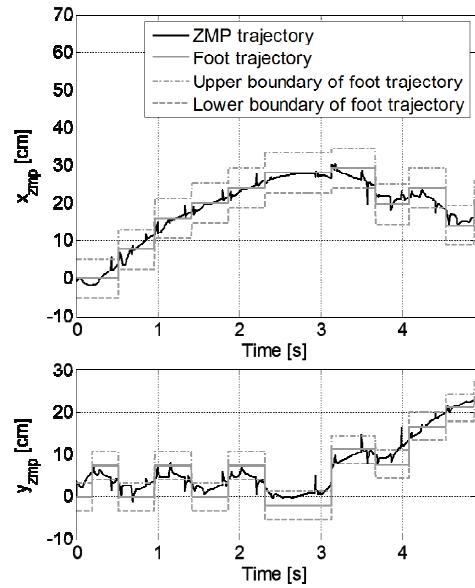


Fig. 13. Measured ZMP trajectories during complex walking simulation: case 1.

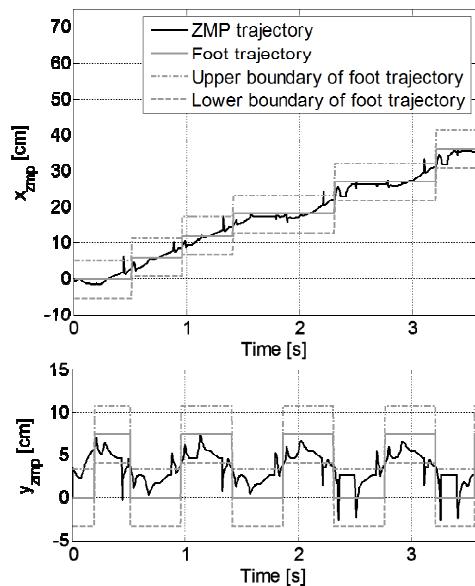
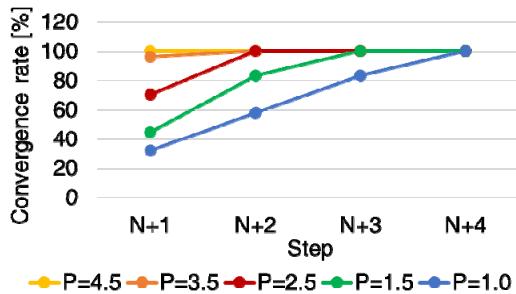


Fig. 14. Measured ZMP trajectories during complex walking simulation: case 2.

Table 4. Sagittal step lengths according to the various allowable ZMP regions.

Step	Sagittal step lengths					
	Ref. length	P=4.5	P=3.5	P=2.5	P=1.5	P=1.0
N	1.0	1.0	1.0	1.0	1.0	1.0
N+1	15.0	15.0	14.4	10.6	6.7	4.8
N+2	15.0	15.0	15.0	15.0	12.5	8.7
N+3	15.0	15.0	15.0	15.0	15.0	12.5
N+4	15.0	15.0	15.0	15.0	15.0	15.0

**Fig. 15.** Convergence rate to the commanded sagittal step length according to allowable ZMP variation. The sagittal step length was changed after the Nth step.

15.0 cm from 1 cm. When $P=4.5$, the commanded step length of 15.0 cm was achieved immediately. Whereas, the sagittal step lengths at the first step were 14.4 cm, 10.6 cm, 6.7 cm and 4.8 cm when $P=3.5$, $P=2.5$, $P=1.5$ and $P=1.0$, respectively. As expected, less allowable ZMP region requires more additional steps to achieve the CS. Fig. 15 represents convergence rates for each allowable ZMP region.

4. Conclusion

The MWPG algorithm makes it possible to move the current walking state and the position and velocity of the CM to the desired values by varying the ZMP while the robot is in the single-support phase. Consequently, the humanoid robot can change its step length, walking period, and direction independently. Even if a navigational command is infeasible, the binary search algorithm substitutes a relaxed one to make it feasible. As a result, a walking pattern can be successfully generated even if a command is infeasible. To show the effectiveness of the algorithm, several walking patterns were generated and examined in the dynamic simulator Webots using an actual dynamic model of the small-sized humanoid platform DARwIn-OP.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of

Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2015R1C1A1A02037641)

References

- [1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A realtime pattern generator for Biped walking," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, vol. 1, pp. 31-37.
- [2] T. Sugihara, Y. Nakamura, and H. Inoue, "Realtime humanoid motion generation through ZMP manipulation based on inverted pendulum control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, vol. 2, pp. 1404-1409.
- [3] Y.-D. Hong and J.-H. Kim, "3-D command state-based modifiable bipedal walking on uneven terrain," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 2, pp. 657-663, Apr. 2013.
- [4] Y.-D. Hong, B.-J. Lee, and J.-H. Kim, "Command state-based modifiable walking pattern generation on an inclined plane in pitch and roll directions for humanoid robots," *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 4, pp. 783-789, Aug. 2011.
- [5] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamagushi, "Integrated motion control for walking, jumping and running on a small bipedal entertainment robot," in *Proc IEEE Int. Conf. robot. Autom.*, Apr. 2004, vol. 4, pp. 3189-3194.
- [6] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," in *Proc. IEEE-RAS/RSJ Int. Conf. Humanoid Robots*, Nov. 2004, vol. 2, pp. 640-655.
- [7] T. Sugihara and Y. Nakamura, "A fast online gait planning with boundary condition realization for humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 305-310.
- [8] B. Ugurlu and A. Kawamura, "Bipedal trajectory generation based on combining inertial forces and intrinsic angular momentum rate changes: Eulerian ZMP resolution," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1406-1415, Dec. 2012.
- [9] KK. Yin, K. Loken, and M. Panne, "Simbicon: simple biped locomotion control," *ACM Trans. Graphics*, vol. 26, no. 3, Jul. 2007.
- [10] KK. Yin, S. Coros, P. Beaudoin, and M. Panne, "Continuation method for adapting simulated skills," *ACM Trans. Graphics*, vol. 27, no. 3, Aug. 2008.
- [11] F. Asano, H. Asoh, M. Morisawa, S. Kajita, and K. Yokoi "Risk evaluation of ground surface using multichannel foot sensors for biped robots," in *Proc. IEEE Int. Symp. Robot. Sensors Environ.*, Oct. 2014, pp. 61-65.
- [12] C. Liu, C. Atkeson, and J. Su, "Biped walking control using a trajectory library," *Robotica*, vol. 31, no. 2, pp. 311-322, Mar. 2013.

- [13] E. Whitman, and C. Atkeson, "Control of a walking biped using a combination of simple policies," in *Proc IEEE/RAS Int. Conf. Humanoid Robot.*, Dec. 2009, pp. 520-527.
- [14] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, pp. 14-19.
- [15] S. Shimmyo, T. Sato, and K. Ohnishi, "Biped walking pattern generation by using preview control based on three-mass model," *IEEE Trans. Ind. Electron.*, vol. 60, no. 11, pp. 5137-5147, Nov. 2013.
- [16] K. Nishiwaki, T. Sugihara, S. Kagami, M. Inaba, H. Inoue, "Online mixture and connection of basic motions for humanoid walking control by footprint specification," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2001, vol. 4, pp. 4110-4115.
- [17] B.-J. Lee, D. Stonier, Y.-D. Kim, J.-K. Yoo, and J.-H. Kim, "Modifiable walking pattern of a humanoid robot by using allowable ZMP variation," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 917-925, Apr. 2008.
- [18] B.-J. Lee and K. I. Kim, "Modifiable walking pattern generation handling infeasible navigational commands for humanoid robots," *J. Elect. Eng. Technol.*, vol. 9, no. 1, pp. 344-351, 2014.
- [19] Q. Huang and Y. Nakamura, "Sensory reflex control for humanoid walking," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 977-984, Oct. 2005.
- [20] Y.-D. Kim, B.-J. Lee, J.-H. Ryu, and J.-H. Kim, "Landing force control for humanoid robot by time domain passivity approach," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1294-1301, Dec. 2007.
- [21] K. Hashimoto, Y. Sugahara, H. Sunazuka, C. Tanaka, A. Ohta, M. Kawase, H. -O. Lim, and A. Takanishi, "Biped landing pattern modification method with nonlinear compliance control," in *Proc. IEEE Int. conf. Robot. Autom.*, May 2006, pp. 1213-1218.
- [22] S. Lim, S.N. Oh, and K.I. Kim, "Balance control for biped walking robots using only zero-moment-point position signal," *Electronics Letters*, vol. 48, no. 1, pp. 19-20, Jan. 2012.
- [23] O. Michel, "Cyberbotics Ltd. WebotsTM: professional mobile robot simulation," *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, pp. 39-42, 2004.



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, Suwon, Korea, where he is currently an Assistant Professor. His current research interests include humanoid robotics, especially in bipedal walking pattern generation, footstep planning, multi/single objective evolutionary optimization, and biologically inspired control.



Ki-Baek Lee received the B.S. and Ph.D. degrees in electrical engineering from KAIST, Daejeon, Korea, in 2005 and 2014, respectively. Since 2014, he has been an Assistant Professor with the Department of Electrical Engineering, Kwangwoon University, Seoul, Korea. He has researched computational intelligence, in particular, in the area of swarm intelligence and multi-objective evolutionary algorithms (MOEAs). His research interests also include a real world application such as on-line multi-objective evolutionary navigation for humanoid robots.



Bumjoo Lee received a B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2002, and M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2004 and 2008, respectively. Since 2012, he has been with the Department of Electrical Engineering at Myongji University, Korea, where he is currently an Assistant Professor. His research interests include humanoid robotics, especially motion planning and control algorithms.