J Electr Eng Technol Vol. 9, No. 1: 344-351, 2014
http://dx.doi.org/10.5370/JEET.2014.9.1.344

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Modifiable Walking Pattern Generation Handling Infeasible Navigational Commands for Humanoid Robots

Bum-Joo Lee* and Kab Il Kim†

Abstract – In order to accomplish complex navigational commands, humanoid robot should be able to modify its walking period, step length and direction independently. In this paper, a novel walking pattern generation algorithm is proposed to satisfy these requirements. Modification of the walking pattern can be considered as a transition between two periodic walking patterns, which follows each navigational command. By assuming the robot as a linear inverted pendulum, the equations of motion between ZMP(Zero Moment Point) and CM(Center of Mass) state is easily derived and analyzed. After navigational command is translated into the desired CM state, corresponding CM motion is generated to achieve the desired state by using simple ZMP functions. Moreover, when the command is not feasible, feasible command is alternated by using binary search algorithm. Subsequently, corresponding CM motion is generated. The effectiveness of the proposed algorithm is verified by computer simulation.

Keywords: Humanoid robot, Bipedal Robot, Walking Pattern, Gait, Locomotion

1. Introduction

In the field of humanoid research, control algorithm of walking pattern plays a significant role. There were approaches based on batch type processed to predesign the walking trajectory and/or control policy. These approaches commonly needed iterative computation to design either entire trajectories or piecewise trajectories over some time interval. As the computing power grows, these approaches became implementable and many successful results were produced [1-3]. To implement the walking pattern in real time, reduced dynamic model such as a single linear inverted pendulum was adopted [4, 5]. In addition, there were some approaches to reduce the complexity of the equation of motion by assuming that the ZMP(Zero Moment Point) trajectory has a specific form [6-9]. In several methods, explicit walking trajectories were not used. These were mainly focused on a sensory feedback algorithm with several intuitive rules [10, 11].

Improving the ability to walk according to the navigational commands is as important as improving the stability. Especially, this is important to deal with navigation within a complex environment [12, 13]. In this paper, to generate walking pattern which satisfies the complex navigational commands, such as walking period, step length and walking direction, a novel algorithm is introduced and resolves the following key points:

- Generation of Periodic Walking Patterns: Ability to generate a periodic walking pattern from the given navigational command.
- Transition between Two Walking Patterns: Ability to modify a walking pattern by the transition from current pattern to the desired one.
- Decision on Feasibility about Navigational Commands: Ability to judge whether or not a navigational command is feasible.

In order to address these subjects, analytical solutions of the equation of motion for the simplified dynamics are utilized. In addition, by adopting the ZMP variation scheme, the proposed algorithm includes the ability to modify the walking pattern without any extra step for adjusting the CM motion. Note that the proposed algorithm is focused on the feed-forward pattern generation. To make robot walk stably, it is essential to adopt some feedback controller sintroduced in [14-17].

The iteration based method such as a preview control scheme utilizes the reference ZMP trajectory as a preview data to give locally optimized walking pattern [3]. In this method, navigational command is implicitly utilized to design the reference ZMP trajectory. Since the proposed method utilizes the pre-defined ZMP functions on the convex hull within the sole, however, the walking patterns are derived from a navigational command explicitly as a closed-form [18]. Consequently, the generated walking patterns satisfy the navigational command exactly if it is feasible. Moreover, it provides a criterion which indicates the navigational command is reachable or not from the current walking state. In this paper, the previous work is further developed to handle the infeasible navigational command by introducing the binary search algorithm.

† Corresponding Author: Dept. of Electrical Engineering, Myongji University, Korea. (kkl@ mju.ac.kr).
* Dept. of Electrical Engineering, Myongji University, Korea. (bjlee@mju.ac.kr).
Received: April 16, 2013; Accepted: August 16, 2013

This paper is organized as follows. In Section 2, walking motion is analyzed in the view of periodic motion and transition between the different periodic motions. Desired goal state for the transition, *desired walking state* is derived in Section 3. The ZMP variation scheme is reviewed in Section 4. Feasibility of the navigational command is considered in Section 5. Section 6 presents example walking pattern using computer simulation. Conclusion is provided in Section 7.

2. Periodic Nature of Walking Pattern

To perform a complex navigational task, the algorithm requires a minimal command set that allows for configuration of the walking pattern. The set of parameters includes single and double supporting times, forward and side step lengths, and walking directions. This instructional set is defined as a *command state*.

Definition 1 Command State(*CS*) is a minimal set to represent 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 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 movement.

Since a *walking* is a periodic motion, it can be represented as a *limit cycle*. When a certain *CS* is specified, the consequent walking pattern is determined. In other words, *walking* according to a given *C* can be analyzed as a movement of state along the periodic orbit which stands for the *walking pattern*. Therefore, in order to handle complex navigational commands, corresponding walking pattern should be figured out. Fig. 1 describes this periodic walking pattern conceptually. Dashed line and dash-dot line represent two different walking patterns, A and B conceptually. When the new navigational command,$\mathbf{c}_\mathrm{B}$, is given during a robot walks according to the current navigational command, $\mathbf{c}_\mathrm{A}$, new walking pattern B is designed according to the $\mathbf{c}_\mathrm{B}$. Subsequently, the robot tries to trans it from the current state, which is depicted as a triangle, to the desired state on the new walking pattern B.

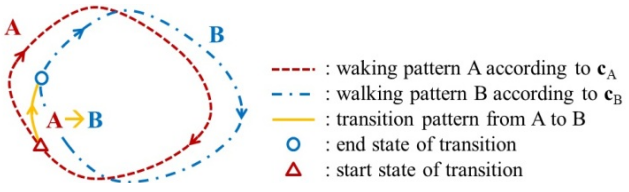


Fig. 1. Transition between two different walking patterns.

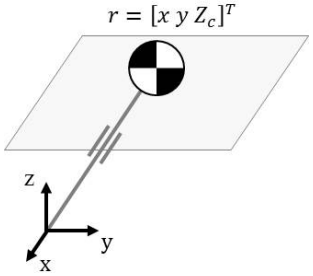


Fig. 2. Linear inverted pendulum model.

In other words, when *CS* is changed into $\mathbf{c}_\mathrm{B}$ from $\mathbf{c}_\mathrm{A}$, a corresponding walking pattern B is resolved. Subsequently, the current state travels to the desired one thereby the walking pattern is modified.

$$\begin{aligned} \mathbf{x} &= \mathbf{A}(\mathrm{t})\mathbf{x}_0 + \mathbf{b}_\mathrm{p}(\mathrm{t}) \\ \mathbf{y} &= \mathbf{A}(\mathrm{t})\mathbf{y}_0 + \mathbf{b}_\mathrm{q}(\mathrm{t}) \end{aligned} \tag{1}$$

where

$$\mathbf{x} \triangleq \begin{bmatrix} x \\ T_c\dot{x} \end{bmatrix},\ \mathbf{y} = \triangleq \begin{bmatrix} y \\ T_c\dot{y} \end{bmatrix}$$

$$\mathbf{A}(\mathrm{t}) \triangleq \begin{bmatrix} c(t) & s(t) \\ s(t) & c(t) \end{bmatrix},\ \mathbf{b}_\mathrm{p}(\mathrm{t}) = -\frac{1}{T_c}\begin{bmatrix} (s * p)(t) \\ (c * p)(t) \end{bmatrix},$$

$$\mathbf{b}_\mathrm{q}(\mathrm{t}) \triangleq -\frac{1}{T_c}\begin{bmatrix} (s * q)(t) \\ (c * q)(t) \end{bmatrix},$$

$c(t)$ and $s(t)$ are abbreviation of $\cosh(t/T_c)$ and $\sinh(t/T_c)$ with a time constant $T_c = \sqrt{Z_c/g}$ and a constant height Z_c, respectively, and $p(\mathrm{t})$ and $q(\mathrm{t})$ represent the ZMP function on the sagittal and the lateral plane, respectively, lastly, $*$ is convolution operator. In spite of conciseness, Eq. (1) characterizes dominant dynamics as well as it separate sagittal and lateral motion.

Note that when the homogeneous solution part in Eq. (1) is only used, namely, the ZMP is assumed that it stays at the center of foot polygon, the CM motion is predetermined by given time, t. Consequently, it is unmodifiable throughout the single support phase. This equivalently means that the humanoid robot cannot accelerate or decelerate its CM. Therfore it can not vary its step length, walking period and direction in the same manner that humans do. The particular solution part, however, relaxes this restriction. By allowing a variation of ZMP over the convex hull on the bounded foot region, it is possible to change the position and the velocity of the CM independently throughout the single support phase.

For the sake of convenience, a state of walking pattern at a given time is defined as a *Walking State*(*WS*). Since in the linear inverted pendulum model, the mass of the robot is considered as a point mass, its walking motion can be represented in terms of the position and the linear velocity. Consequently, the position and the velocity of the CM become the *WS* as follows:

Definition 2 Walking State (*WS*) for the linear inverted pendulum represents the CM state as follows:

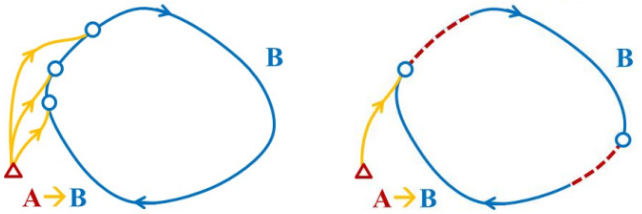
$$\mathbf{x} \triangleq [x\ T_c\dot{x}]^T \text{ for the sagittal motion,}$$
$$\mathbf{y} \triangleq [y\ T_c\dot{y}]^T \quad \text{for the lateral motion.}$$

Note that the velocity term is weighted with the time constant, T_c. From now on, walking motion is analyzed in terms of the *WS*.

3. Desired Walking State

Modification of a walking pattern can be characterized into two parts: generation of another periodic walking pattern which satisfies a given new *CS* and transition of *WS* from a current state to a desired state on the new trajectory.

Fig. 3 represents two different walking patterns and transitions between them, where the limit cycle means a new walking pattern which is generated by the new *CS*. The triangle and the circle represent current and *desired WS*, respectively. Lastly, solid line from the triangle to the circle indicates transitional walking pattern. As shown in Fig. 3 (a), transition from the current state can be accomplished in various ways. In other words, a lot of*desired WS* and transitional walking patterns are possible. Therefore it arise an optimization issue. In practical manner, *desired WS* is restricted to a final state of single support phase which is depicted as a circle on a solid-line for the single support phase. While a dashed-line means double support phase on walking pattern B as shown in Fig. 3 (b).



(a) (b)

Fig. 3. Transitions of *WS*. (a) Various possible transitions. (b) Proposed transition strategy.

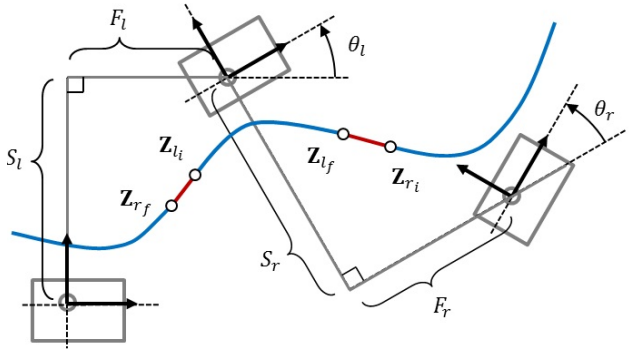


Fig. 4. Foot print of steady walking.

Moreover, it is assumed that the ZMP variation carries out only during the transition. Consequently, *WS* on the limit cycle is characterized by only the homogeneous solution part of Eq. (1). Using this strategy, the *desired WS* is uniquely determined from a given *CS* and a *current WS* as shown in Fig. 4 which describes a periodic walking pattern according to the *CS*.

In the figure, $\mathbf{Z} \triangleq [\mathbf{xy}]$ and subscripts, $l_{i(f)}$ and $r_{i(f)}$, mean left initial(final) and right initial(final) state of the single support, respectively. Lastly, θ implies walking direction.From Eq. (1), motions of the single support phase are derived as follows:

$$\begin{aligned}\mathbf{Z}_{l_f} &= \mathbf{A}_{T_{sl}}\mathbf{Z}_l \\ \mathbf{Z}_{r_f} &= \mathbf{A}_{T_{sr}}\mathbf{Z}_{r_i}\end{aligned} \tag{2}$$

As mentioned above, during the periodic motion ZMP does not vary and stays at the center of foot polygon. Consequently, the motion of equation is characterized with only homogeneous part. Also, double support motions which are controlled with constant velocity are obtained as follows:

$$\begin{aligned}\mathbf{Z}_{l_i} &= \left(\mathbf{U}_{T_{dr}}\mathbf{Z}_{r_f} - \mathbf{D}_l\right)\mathbf{R}_{\theta_l} \\ \mathbf{Z}_{ri} &= \left(\mathbf{U}_{T_{dl}}\mathbf{Z}_{l_f} - \mathbf{D}_r\right)\mathbf{R}_{\theta_r}\end{aligned} \tag{3}$$

where

$$\mathbf{U}_{\mathrm{t}} \triangleq \begin{bmatrix}1 & t/T_c \\ 0 & 1\end{bmatrix},\ \mathbf{D}_{\mathrm{l(r)}} \triangleq \begin{bmatrix}F_{l(r)} & S_{l(r)} \\ 0 & 0\end{bmatrix},$$
$$\mathbf{R}_{\theta} \triangleq \begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}$$

By eliminating the *initial WS* from Eq. (2) and (3), homogeneous solutions of the steady motion are given as follows:

$$\begin{aligned}\mathbf{Z}_{l_f} &= \mathbf{A}_{T_{sl}}\mathbf{U}_{T_{dr}}\mathbf{Z}_{r_f}\mathbf{R}_{\theta_l} - \mathbf{A}_{T_{sl}}\mathbf{D}_l\mathbf{R}_{\theta_l} \\ \mathbf{Z}_{r_f} &= \mathbf{A}_{T_{sr}}\mathbf{U}_{T_{dl}}\mathbf{Z}_{l_f}\mathbf{R}_{\theta_r} - \mathbf{A}_{T_{sr}}\mathbf{D}_r\mathbf{R}_{\theta_r}\end{aligned} \tag{4}$$

Using the *Kronecker Product*, Eq. (4) is transformed into the familiar linear equation form as follows:

$$\begin{aligned}\boldsymbol{\zeta}_{l_f} &= \mathbf{A}_r^*\boldsymbol{\zeta}_{r_f} - \mathbf{b}_r^* \\ \boldsymbol{\zeta}_{r_f} &= \mathbf{A}_l^*\boldsymbol{\zeta}_{l_f} - \mathbf{b}_l^*\end{aligned} \tag{5}$$

where

$$\boldsymbol{\zeta} \triangleq \mathrm{vec}(\mathbf{Z})$$
$$\mathbf{A}_{r(l)}^* \triangleq \mathbf{R}_{\theta_{r(l)}}^T \otimes \left(\mathbf{A}_{T_{sr(l)}}\mathbf{U}_{T_{dl(r)}}\right)$$
$$\mathbf{b}_{r(l)}^* \triangleq \mathrm{vec}\left(\mathbf{A}_{l(r)}\mathbf{D}_{l(r)}\mathbf{R}_{\theta_{l(r)}}\right)$$

Finally, solving Eq. (5) for the final *WS*s, $\boldsymbol{\zeta}_{l_f}$ and $\boldsymbol{\zeta}_{r_f}$,

*desired WS*s are given as follows:

$$\begin{aligned}\boldsymbol{\zeta}_{l_f} &= (\mathbf{A}_r^* \mathbf{A}_l^* - \mathbf{I}_{4\times4})^{-1}(\mathbf{A}_r^* \mathbf{b}_l^* + \mathbf{b}_r^*) \\ \boldsymbol{\zeta}_{r_f} &= (\mathbf{A}_l^* \mathbf{A}_r^* - \mathbf{I}_{4\times4})^{-1}(\mathbf{A}_l^* \mathbf{b}_r^* + \mathbf{b}_l^*)\end{aligned} \tag{6}$$

Eq. (6) expresses the mapping relationship between the *CS* and the *desired WS*. Note that the information of the *CS* is involved in the matrix $\mathbf{A}^*$ and $\mathbf{b}^*$. Whenever the *CS* is changed, it translates into the *desired WS* form. Subsequently, $\boldsymbol{\zeta}_{lf}$ or $\boldsymbol{\zeta}_{rf}$ becomes the *desired WS* for the left or right support phase, respectively.

4. ZMP Variation Scheme

After the *desired WS* is calculated, it is necessary to adjust the ZMP trajectory to transfer the *current WS* to the desired one. Via the variation of the ZMP with closed-form function, it is possible to modify the walking pattern as intended by the *CS*. In a practical manner, a constant function(Fig. 5 (a)) and a step function(Fig. 5 (b)) are proposed for sagittal motion and lateral motion, respectively. Detailed analysis of these two ZMP functions is covered in the previous work [18].

The control parameters (T , P , T_{sw} and Q) which characterize the proposed ZMP functions are now solved to ensure that the desired *WS* is attainable.

4.1 Sagittal motion

By substituting $t = T$ and $p(t) = P$ into Eq. (1), following solutions are obtained:

$$\begin{aligned}T &= T_c \ln\left(\frac{(v_d+v_c)+(x_d-x_c)}{(v_d+v_c)-(x_d-x_c)}\right) \\ P &= \frac{(x_d^2-x_c^2)-(v_d^2-v_c^2)}{2(x_d-x_c)}\end{aligned} \tag{7}$$

where the subscripts c and d and represent *current* and *desired* state, respectively. Note that T represents the requiredtransition time from the *current WS* to the *desired WS*.

4.2 Lateral motion

Since the homogeneous part of Eq. (1) for the lateral motion is determined by the *current WS* and the transition time T, it is only necessary to consider the particular part. Similarly to the sagittal motion, by letting

$$q(\mathrm{t}) = \begin{cases} Q, & (0 \le t \le T_{sw}) \\ -Q, & (T_{sw} \le t \le T) \end{cases}$$

the particular solution of the lateral equation of motion is derived. Solving this for the unknowns, T_{sw} and Q, gives

$$\begin{aligned}T_{sw} &= T - T_c\, ln(h) \\ Q &= \frac{y_p}{h+h^{-1}-(1+c(T))}\end{aligned} \tag{8}$$

where

$$h = \begin{cases} \frac{\gamma-\sqrt{\gamma^2+4\alpha\beta}}{2\alpha}, & (\delta < 0) \\ 1, & (\delta = 0) \\ \frac{\gamma+\sqrt{\gamma^2+4\alpha\beta}}{2\alpha}, & (\delta > 0) \end{cases}$$

$$\alpha = y_p - T\_cw_p,\ \beta = y_p + T\_cw_p$$

$$\gamma = y_p s(T) - T\_cw\_p\big(1 + c(T)\big),$$

$$\delta = y_p s + T_c w_p\big(1 - c(T)\big)$$

Using Eq. (7) and Eq. (8), control parameters are directly obtained from the current and the *desired WS*. Subsequently, *WS* is transited by the ZMP variation.

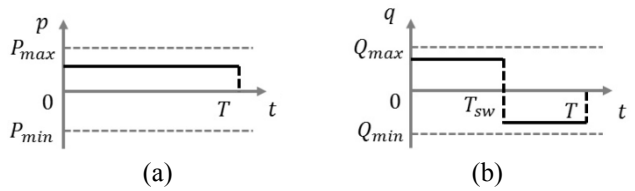


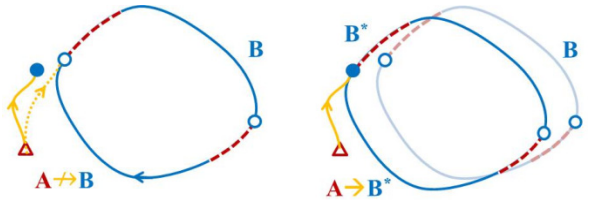
Fig. 5. Proposed ZMP functions. (a) Constant function for the sagittal motion. (b) Step function for the lateral motion.

5. Modification of Infeasible Command State and Desired Walking State

As explained in the previous section, a *desired WS* is achieved by the ZMP variation. Due to the several kinematical and/or dynamical constraints, however, sometimes *desired WS* is not attainable. For instance, when the magnitude of the ZMP function for the sagittal and/or lateral motion is larger or smaller than the maximum or the minimum limit, there is no way to achieve the *desired WS*. In this case, the *WS* should be substituted to the nearest one.This is illustrated in Fig. 6, where the empty circle means an infeasible *desired WS* whereas the solid circle means a feasible *desired WS* substituted from the original one.

Since the substituted *WS* is not on the desired walking pattern B (see Fig. 6 (a).), it is necessary to re-compute a new walking pattern B* through the new *WS* (see Fig. 6 (b).). To do so, however, modification of the original *CS* has to be preceded because the walking pattern is dependent on the *CS*. Subsequently, the *current WS* can travel to the replaced feasible *WS* on the new walking pattern.

In this paper, infeasible *CS* and *desired WS* are simultaneously modified by using *Binary Search Algorithm* as shown in Algorithm 1, where subscripts p and d represent *previous* and *desired* state, respectively, and m



(a) (b)

Fig. 6. Modification of *desired WS* corresponding to infeasible *CS*. (a) Infeasibility of *desired WS*. (b) New walking pattern from the modified *CS*

Algorithm 1 Feasible CS and corresponding desired WS

/* initialize variables */

$\mathbf{c}_0 \leftarrow \mathbf{c}_p$;
$\mathbf{c}_1 \leftarrow \mathbf{c}_d$;
$\mathbf{c}_f \leftarrow \mathbf{c}_p$;
$\boldsymbol{\zeta}_f \leftarrow f(\mathbf{c}_f)$
$n \leftarrow 0$;

/* start binary search algorithm */

while $n < N$ **do**
$\mathbf{c}_m \leftarrow (\mathbf{c}_0 + \mathbf{c}_1)/2$;
$\boldsymbol{\zeta}_m \leftarrow f(\mathbf{c}_m)$;
if $\boldsymbol{\zeta}_m$ is *feasible* **then**
$\mathbf{c}_0 \leftarrow \mathbf{c}_m$;
/* update *desired WS* and *CS* */
$\mathbf{c}_f \leftarrow \mathbf{c}_m$;
$\boldsymbol{\zeta}_f \leftarrow \boldsymbol{\zeta}_m$;
else
$\mathbf{c}_1 \leftarrow \mathbf{c}_m$;
end if
$n \leftarrow n + 1$;
end while

and f represent *modified* and *feasible* state, respectively, lastly, $f(\cdot)$ means a function gives a *desired WS* from a *CS* by Eq. (6). After initialization, *CS* is updated as a mean value between previous and desired *CS*. Note that the *modified CS* is always feasible because the current *WS* is on the walking pattern generated by the previous *CS*. Subsequently, if the modified *desired WS* is feasible then the modified *desired WS* moves to the desired one, else it moves to the previous one until the iteration reaches to the maximum. This *Binary Search Algorithm* is simple and intuitive. Moreover it is sufficiently fast to execute in real time.

6. Example Walking Pattern

The proposed algorithm was simulated to verify the effectiveness with several example walking patterns including *forward walking*, *backward walking*, *sideways walking*, and *turning motion*. Parameters used in simulation are shown in Table 1, where Z_c means constant CM height in the linear inverted pendulum model (see Fig. 2.) and,

Table 1. Simulation parameters.

Z_c	P_{max}	P_{min}	Q_{max}	Q_{min}
0.6	0.1	−0.05	0.05	−0.05

$P_{max(min)}$ and $Q_{\max(min)}$ represent maximum(minimum) ZMP variation region for sagittal and lateral motion, respectively. Note that all length units are given in meters.

6.1 Forward walking

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

Initial *CS*,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 2^{nd} step,
$\mathbf{c} = [0.3\ 0.3\ 0.05\ 0.05\ 0.4\ 0.4\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 5^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 6^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.0\ 0.0\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$

where time and length units are given in seconds and meters, respectively, and angle unit is given in degree. Note that the *CS* after the k^{th} step is not delivered until the double support phase of the k^{th} step is finished. In other words, the proposed algorithm does not use preview data. In the all example walking patterns during this simulation, robot was commanded to stop after accomplishing the final *CS*.

Fig. 7 shows the walking pattern generated by the proposed method. The solid curve and the rectangle indicate the CM trajectory and the foot polygon, respectively. The circle and the number mean the center of foot and the step number, respectively. The initial foot positions are $(0, 0)$ for the right foot and $(0, 0.2)$ for the left foot, respectively. As the *CS* list intended, robot lengthened its stride after the 2^{nd} step and shortened after the 5^{th} step. Although the robot was commanded to lengthen its step length as 0.4m after the 2^{nd} step, however, the step length is increased smoothly for $0.3108(= 0.7108 - 0.4)$ m at the 3^{rd} step and $0.3792(= 1.09 - 0.7108)$ m at the 4^{th} step, respectively. It finally achieved the commanded step length, $0.4(= 1.49 - 1.09)$ m, at the 5^{th} step. Also it can be seen that similar result is observed at the 6^{th} step. These

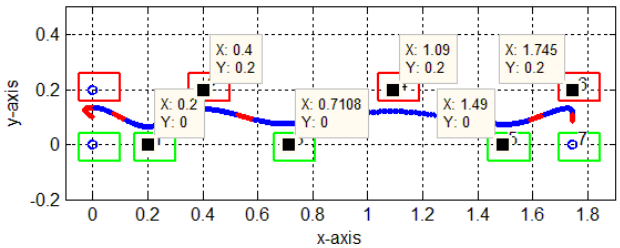


Fig. 7. Example pattern (forward walking).

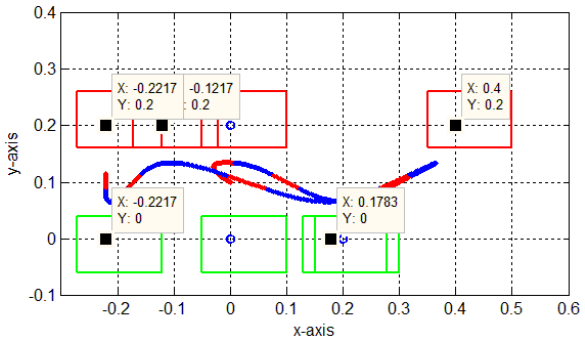


Fig. 8. Example pattern (backward walking).

were caused by that the infeasible *CS*s delivered after the 2^{nd} and the 5^{th} steps were substituted for the relaxed *CS* which is thereby feasible.

6.2 Backward walking

Backward walking patterns were realized from the following *CS* list:

Initial CS,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°0.0°]^T$
After the 2^{nd} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ -0.3\ -0.3\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 4^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ -0.1\ -0.1\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 5^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.0\ 0.0\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$

Fig. 8 shows the backward walking pattern generated from the *CS* list. According to the *CS* list, walking patterns were generated to walk backward and stop after the 5^{th} step. Similar with the forward walking case, although the commanded backward step length was 0.3 m after the 2^{nd} step, it is not satisfied until the 4^{th} step. It is finally achieved for the step length at the 4^{th} step.

6.3 Sideways walking

Sideways walking patterns were realized from the following *CS* list:

Initial CS,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 2^{nd} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.1\ 0.1\ 0.2\ -0.4\ 0.0°\ 0.0°]^T$
After the 5^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 6^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.0\ 0.0\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$

Fig. 9 shows the sideways walking pattern. According to the *CS* list, side step lengths are designed as different as

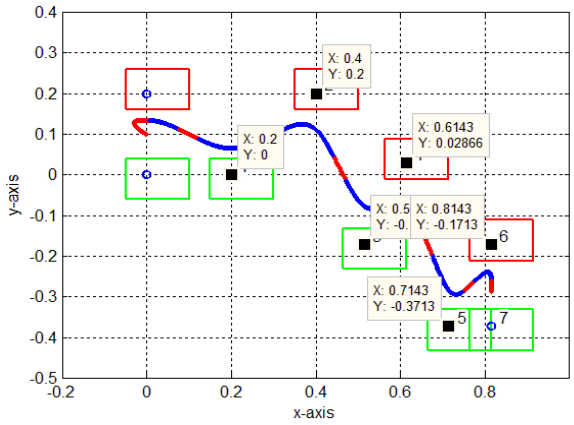


Fig. 9. Example pattern (sidewayswalking).

0.2 m and -0.4 m for the left and the right step, respectively. Via the 3^{rd} step, which has a small difference between the commanded and the generated side step lengths due to the relaxation of the infeasible *CS*, subsequent footsteps were followed the given *CS*s accurately.

6.4 Turning motion

Turning motions were realized from the following *CS* list:

Initial CS,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 2^{nd} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ -60.0°]^T$
After the 5^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 6^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.0\ 0.0\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$

Fig. 10 shows the turning motion. After the 2^{nd} step, it started to turn 30° to the right and stopped after the 6^{th} step.

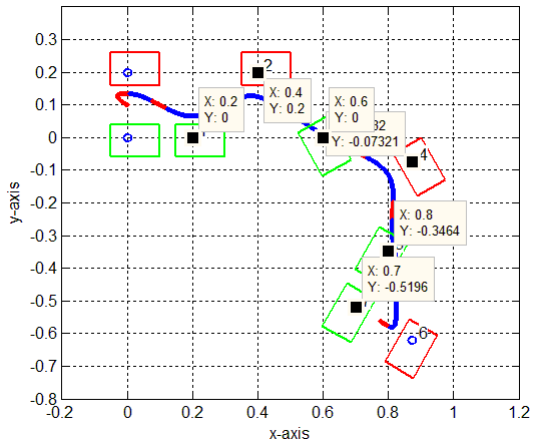


Fig. 10. Example walking (turning motion).

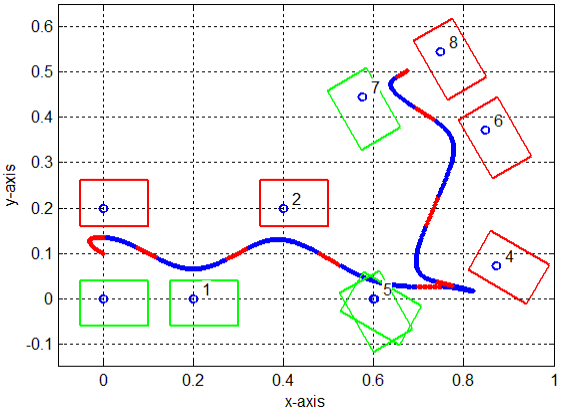


Fig. 11. Example pattern (complex walking).

6.5 Complex walking

Complex walking patterns including forward, backward, sideways and turning motion were realized from the following *CS* list:

Initial CS,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°0.0°]^T$
After the 2^{nd} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.2\ 0.2\ 0.2\ -0.2\ 0.0°\ -30.0°]^T$
After the 4^{nd} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ -0.2\ -0.2\ 0.4\ -0.2\ 0.0°\ -30.0°]^T$
After the 7^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ -0.2\ -0.2\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$
After the 8^{th} step,
$\mathbf{c} = [0.4\ 0.4\ 0.1\ 0.1\ 0.0\ 0.0\ 0.2\ -0.2\ 0.0°\ 0.0°]^T$

Fig. 11 shows the generated walking pattern. After the 2^{nd} step, the wakling direction was changed to the right. After the 4^{th} step, it started to walk backward with different side step length. Subsequently, it stopped after the 8^{th} step.

7. Conclusion

The modifiable walking pattern generation algorithm was proposed. By varying the ZMP(Zero Moment Point) while in a single support phase, it is possible to move the current walking state, the position and the velocity of the CM(Center of Mass), to the desired one. Consequently, the humanoid robot could change its step length, walking period and direction, independently. In addition, when the navigational command is infeasible, it is substituted for the relaxed one by the binary search algorithm and thereby becomes feasible. As a result, walking pattern could be successfully generated even though the command was infeasible. Since the proposed method utilizes the closed-form functions and simple search algorithm to generate walking pattern, it is possible to compute in real time.

Acknowledgement

This work was supported by Unmanned Technology Research Centre (UTRC).

References

[1] C. Liu, C. Atkeson, and J. Su, "Biped walking control using a trajectory library," *robotica*, May 2012 (published online).

[2] E. Whitman, and C. Atkeson, "Control of a walking biped using a combination of simple policies," in *IEEE/RAS Int. Conf. Humanoid Robot.*, Paris, France, Dec. 2009, pp. 520-527.

[3] 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.*, Taipei, Taiwan, Sep. 2003, pp. 14-19.

[4] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, andH. Hirukawa, 'A realtime pattern generator for Biped walking', *Proc.IEEE Int. Conf. Robot. Autom.*,Washington, DC, vol. 1, pp. 31-37, May 2002.

[5] 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.*, Washington, DC, May 2002, vol. 2, pp. 1404-1409.

[6] 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 *IEEE Int. Conf. robot. Autom.*, New Orleans, LA, Apr. 2004, vol. 4, pp. 3189-3194.

[7] 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*, Los Angeles, CA, Nov. 2004, vol 2, pp. 640-655.

[8] T. Sugihara and Y. Nakamura, "A fast online gait planning with boundary condition relazation for humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 305-310.

[9] R. Kurazume, T. Hasegawa, and K. Yoneda, "The sway compendation trajectory for a biped robot," in *Proc. IEEE int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 2003, vol. 1, pp. 925-931.

[10] KK. Yin, K. Loken, and M. Panne, "Simbicon: simple biped locomotion control," *ACM Trans. on Graphics*, vol. 26, no. 3, Jul. 2007.

[11] KK. Yin, S. Coros, P. Beaudoin, and M. Panne, "Continuation method for adapting simulated skills," *ACM Trans. on Graphics*, vol. 27, no. 3, Aug. 2008.

[12] P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J.

Kuffner, and T. Kanade, "Online environment reconbstruction for biped navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, May 2006, pp. 3089-3094.

[13] J. Chestnutt, P. Michel, N. Nishiwaki, J. Kuffner, and S. Kagami, "An intelligent joystic for biped control," in *Proc. IEEE Int. conf. Robot. Autom.*, Orlando, FL, May 2006, pp. 860-865.

[14] Q. Huang and Y. Nakamura, "Sensory reflex control for humanoid walking," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 977-984, Oct. 2005.

[15] 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.

[16] K. Hashimoto, Y. Sugahara, H. Sunazuka, C. Tanaka, A. Ohta, M. Kawase, H. -O. Lim, and A. Takanish, "Biped landing pattern modification method with nonlinear compliance control," in *Proc. IEEE Int. conf. Robot. Autom.*, Orlando, FL, May 2006, pp. 1213-1218.

[17] S. Lim, S.N. Oh and K.I. Kim,: 'Balance control for biped walking robotsusing only zero-moment-point position signal', *Electronics Letters*, vol. 48, no. 1, pp. 19-20, Jan. 2012.

[18] Lee, B.-J., Stonier, D., Kim, Y.-D., Yoo, J.-K., and Kim, J.-H., 'Modifiable Walking Pattern of a Humanoid Robot by Using Allowable ZMP Variation', *IEEE Trans. on Robotics*, vol. 24, no. 4, pp. 917-925, Apr. 2008.

Bum-Joo Lee received the B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2002, and the 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, Myongji University, Korea, where he is currently an Assistant Professor. His research interests include the areas of Humanoid Robotics, especially in motion planning and control algorithm.

Kab Il Kim graduated Seoul National University (SNU) for B.S. in 1979, Korea Advanced Institute of Science and Technology (KAIST) for M.S. in 1981, and Clemson University in South Carolina (USA) for Ph.D. in 1990, respectively. Since 1991, he has been with the Department of Electrical Engineering, Myongji University, Korea, where he is currently a Professor. He studied Two Arm Coordination, Load Distribution, and Humanoid Robotics as a main research topic. On the practical aspect, he is currently interested in the area of Field Robotis and Military robotics.