2013 13th IEEE-RAS International Conference on
Humanoid Robots (Humanoids).
October 15 - 17, 2013. Atlanta, GA

Velocity Tracking Algorithm in Forward Walking using Estimated ZMP by Whole-Body Control Framework

Yisoo Lee and Jaeheung Park

Abstract— **This paper presents a forward walking strategy for humanoid robots using a whole-body control framework. Implementation of walking requires the coordination of movements between the feet and the center of mass (COM) of the system. In addition, the condition of preventing falling needs to be considered at the planning stage and later the robot is controlled to meet this condition upon execution. In this paper, assuming that the planned velocity of the COM and the corresponding foot trajectories are given, generating the desired COM position and velocity by estimating the ZMP of the foot is proposed to account for the stability condition. This strategy provides a way to implement walking without the explicit planning of the COM in the forward direction. The proposed approach is implemented and verified in a simulation of a humanoid robot.**

I. INTRODUCTION

Controlling walking motion is one of the most important and fundamental issues in humanoid robots. The planning and control of the COM during walking is especially important because it is not only directly related to locomotion itself but also related to the stability of the robot. Therefore, the COM must be planned to move the robot from one place to another while meeting the stability condition, i.e., not to fall.

Many walking pattern-generation algorithms have been proposed based on the ZMP (Zero Moment Point) concept because the ZMP represents the criteria of the dynamic balancing of biped robots [1] [2]. In earlier work [3] [4], the ZMP is calculated based on the dynamics of each link of the robot for biped walking. To simplify the calculation for generating a walking pattern, the Linear Inverted Pendulum Model (LIPM) was proposed, in which a robot is assumed to be a single mass and a walking pattern is generated in real time [5]. However most methods based on the LIPM are limited in their ability to modify foot placement to a specified location. To solve this problem, a preview control algorithm with a ZMP based LIPM was proposed and its efficiency was proved with experiments [6]. It remains limited in terms of its ability to handle the ZMP when a sudden disturbance appears or the foot location needs to be changed abruptly, as preview control requires a future ZMP plan. Auxiliary ZMP control [7] and a method for changing the ZMP trajectory within a permissible region [8] were introduced to overcome this problem. The LIPM based method without the use of a preview controller was also developed to use the ZMP. Modifiable walking pattern generation based on an allowable ZMP variation was also proposed [9]. A ZMP reference generation algorithm was introduced, showing that a moving ZMP is more energy-efficient than a fixed ZMP [10].

However, ZMP error always exists because the LIPM is a simplified model which does not reflect the whole-body dynamics of a robot. To reduce ZMP error, the LIPM which considers changes in the angular momentum was proposed. It successfully decreased the ZMP error and torso angle variations about the pitch and roll axes [11]. Another way to prevent unstable situations caused by ZMP error involves the use of stabilizers with Force/Moment feedback [12] [13]. In two studies [6] [14], a dynamic filter which calculates multi-body dynamics is used to estimate ZMP error, which is compensated for by changing joint trajectories.

In this paper, we propose a velocity tracking algorithm with a variable ZMP in forward walking using the estimated ZMP that is computed by a multi-body dynamics model of a robot. The goal is to implement forward walking without explicit COM trajectory generation by utilizing the available ZMP within a whole-body control framework. First, the torque command is computed when the desired velocity and position of the COM are commanded, which are determined based on the planned velocity. The ZMP is then estimated by the whole-body control framework [15]. If the estimated ZMP violates the ZMP stability condition, the desired velocity is then modified. The difference between the planned velocity and the modified desired velocity of the COM is integrated to check if the desired velocity on average is below the planned velocity of the COM. Depending on whether the integrated value is negative or positive, higher or lower velocities are commanded whenever the ZMP has margins within the foot boundary. This proposed approach differs from most other approaches because there is no explicit planning of COM trajectories.

This paper is organized as follows. Explanation of the whole-body control framework for humanoid robots and ZMP estimation method are given in Section II. Section III presents a strategy for generating COM trajectories and walking motions in the forward direction based on the planned velocity and on estimations of the ZMP. The proposed strategy for walking is then implemented in a simulation using a humanoid robot model. The simulation results are presented in Section IV. The conclusion is given in Section V.

Yisoo Lee and Jaeheung Park is with Graduate School of Convergence Science and Technology, Advanced Institutes of Convergence Technology, Seoul National University, Republic of Korea. E-mails: {howcan11, park73}@snu.ac.kr. Jaeheung Park is the corresponding author.

II. ZMP ESTIMATION IN WHOLE-BODY CONTROL FRAMEWORK

A. Contact-Consistent Whole-Body Control Framework

The contact-consistent whole-body control framework provides a control structure in which multiple tasks are defined and executed using the full dynamics of a robot in multiple contact situations [12]. In this paper, a biped robot is controlled to track planned trajectories using this control framework. The tasks can be defined as a vector of the corresponding coordinates such as $X = [X_{COM}^T,\ X_{LeftFoot}^T,\ \cdots]^T$. Then, the task Jacobian is similarly constructed by stacking the corresponding Jacobians. The control force for the tasks can be composed with the following dynamics in the task space:

$$\Lambda(q)\ddot{X} + \mu(q,\dot{q}) + p(q) = F \tag{1}$$

where

$$\begin{aligned} N_c &= I - J_c^T \bar{J}_c^T && (2)\\ \Lambda(q) &= [JA^{-1}N_cJ^T]^{-1} && (3)\\ \bar{J}^T &= \Lambda JA^{-1}N_c && (4)\\ \mu(q,\dot{q}) &= \bar{J}^T b(q,\dot{q}) - \Lambda\dot{J}\dot{q} + \Lambda JA^{-1}J_c^T\Lambda_c\dot{J}_c\dot{q} && (5)\\ p(q) &= \bar{J}^T g(q). && (6) \end{aligned}$$

Here, $\Lambda(q)$ is the inertia matrix in the operational space, $\mu(q,\dot{q})$ is the vector of the Coriolis and centrifugal forces at the operational space, and $p(q)$ is the vector of the gravity forces at the operational space. The matrix A is the inertia matrix in the joint space, $b(q,\dot{q})$ is the Coriolis and centrifugal vector, and $g(q)$ is the gravity vector. The contact Jacobian J_c is defined as $\vartheta_c = J_c\dot{q}$, where ϑ_c is the linear and angular velocity of the contact point and q is the vector of the joint angles. For the control force F in the operational space, the control torque Γ can be computed by

$$\begin{aligned} \Gamma &= (J^k)^T F \\ &= (J^k)^T\Lambda\{f^* + \mu + p\}. \end{aligned} \tag{7}$$

where $(J^k)^T$ is a modified task Jacobian accounting for contacts and f^* is the desired acceleration. For PD control,

$$f^* = k_p(x_d - x) + k_d(\dot{x}_d - \dot{x}), \tag{8}$$

where x_d is the vector of the desired position, the term $\dot{x}_d$ is the vector of the desired velocity, and the values k_p and k_d are the proportional and derivative gain. One earlier study contains details pertaining to the whole-body control framework [12].

B. ZMP Estimation

When the robot is controlled by the whole-body control framework described in Section II-A, the expected contact forces and moments can be calculated in real time. The ZMP is then estimated using these values. That is, given the control torque Γ computed for the tasks, the expected contact forces and moments are obtained by

$$f_c = \bar{J}_c^T\Gamma - \mu_c - p_c, \tag{9}$$

where

$$\begin{aligned} \Lambda_c &= (J_cA^{-1}J_c^T)^{-1} && (10)\\ \mu_c &= \Lambda_c\{J_cA^{-1}b(q,\dot{q}) - \dot{J}_c\dot{q}\} && (11)\\ p_c &= \Lambda_cJ_cA^{-1}g(q) && (12)\\ \bar{J}_c^T &= \Lambda_cJ_cA^{-1}. && (13) \end{aligned}$$

The vector f_c is the expected contact force and moment vector. The matrix $\bar{J}_c$ is the dynamically consistent inverse of J_c, the vector μ_c is the projection of the Coriolis and centrifugal vector on the contact space, and the vector p_c is the projection of the gravity vector on the contact space. When there is single-plane contact at the supporting foot where the contact force vector $f_c = [F_x\ F_y\ F_z\ M_x\ M_y\ M_z]^T$, the ZMP can be estimated using these contact forces and moments.

$$\begin{aligned} r_{x,est} &= -\frac{M_y}{F_z}, && (14)\\ r_{y,est} &= \frac{M_x}{F_z}, && (15) \end{aligned}$$

where $r_{x,est}$ and $r_{y,est}$ are the estimated ZMP values in the x and y directions, respectively. The ZMP is estimated with the desired torque values, which are computed for the execution of the desired tasks. In the following sections, the estimated ZMP is used to modify the desired task commands and torques before actually sending the commanded torques to the robot to allow the actually commanded torque to meet the stability condition.

III. VELOCITY TRACKING IN FORWARD WALKING USING ZMP ESTIMATION

A. Walking in Forward Direction

Walking in a forward direction, i.e., in a sagittal direction, is implemented given the planned velocity and foot step length. First, given these parameters, the foot trajectories are designed in both the sagittal and vertical directions. The initial COM trajectory is generated from the given planned velocity. Then, these tasks are commanded to the whole-body control framework. The execution of these tasks, however, can violate the condition that the ZMP must remain within the supporting polygon. Whether or not this violation occurs can be checked using the ZMP estimation explained in the previous section.

Once the ZMP is expected to go over the limits, the ZMP can be modified to stay within the boundary by either initiating a null space control strategy or by changing the tasks of the COM or the swinging foot. In this paper, changing the COM trajectory for the ZMP to remain within the boundaries is investigated to achieve the desired speed of walking. When the estimated ZMP is ahead of the front boundary of the supporting foot, the desired ZMP can be modified and thus pulled into the boundary. This will increase the acceleration of the COM in the forward direction while maintaining the foot trajectories as they were. Alternatively, when the estimated ZMP is below the boundary of the supporting foot, the COM will be decelerated by pushing the ZMP location forward. Although this algorithm can prevent the robot from

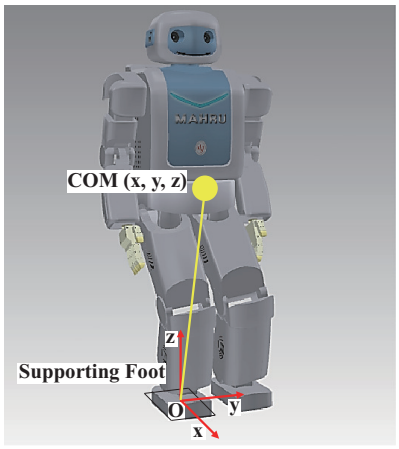

Fig. 1. **The coordinate for the walking on the biped robot**. The COM is described based on the frame, which is at the center of the supporting foot.

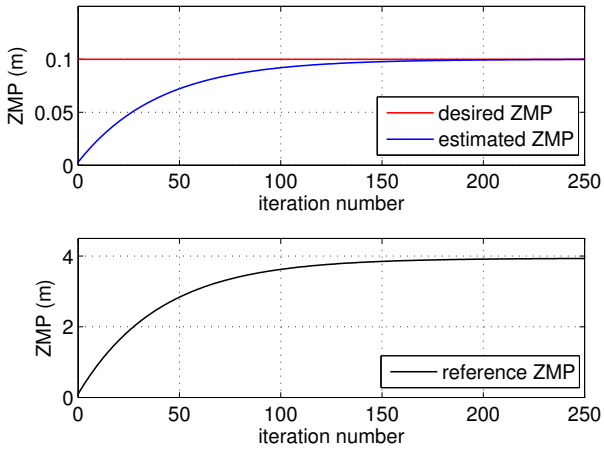

Fig. 2. **Iteration result of ZMP**. With an increase in the number of iterations, the estimated ZMP converges to the desired ZMP.

falling, it does not guarantee the robot to follow the desired velocity. Therefore, the proposed strategy is to keep track of the difference between the desired and planned velocities, and to increase or decrease the desired velocity whenever the ZMP allows.

As part of this process, it is necessary to compute the desired position and velocity of the COM in the next servo cycle to achieve the modified desired ZMP. The LIPM model is used to approximate the relationship between the COM and ZMP. The whole-body control algorithm is then applied to determine the exact value of the desired COM position and velocity by iteration.

B. COM Trajectory Modification using ZMP Estimation and LIPM

The LIPM provides trajectories for the position and velocity of a COM reflecting a reference ZMP [9]. It can be expressed as

$$\begin{bmatrix} x_d \\ T_c v_d \end{bmatrix} = \begin{bmatrix} \cosh(\frac{\Delta t}{T_c}) & \sinh(\frac{\Delta t}{T_c}) \\ \sinh(\frac{\Delta t}{T_c}) & \cosh(\frac{\Delta t}{T_c}) \end{bmatrix} \begin{bmatrix} x \\ T_c v \end{bmatrix} - \frac{1}{T_c} \begin{bmatrix} (\cosh(\frac{\Delta t}{T_c}) - 1) r_{x,ref} \\ \sinh(\frac{\Delta t}{T_c}) r_{x,ref} \end{bmatrix}, \quad (16)$$

where Δt is the control period of the robot system, x_d and v_d are respectively the desired COM position and velocity in the sagittal direction, x and v are likewise the current COM position and velocity in the sagittal direction and $r_{x,ref}$ is the reference ZMP for the sagittal direction of the LIPM. The time constant $T_c = \sqrt{z_c/g}$, where z_c is the height of the COM and g denotes the degree of acceleration by gravity.

First, the reference ZMP is selected as the desired ZMP($r_{x,d}$). The desired position and velocity of the COM can be computed using the model. The estimated ZMP can then be calculated by commanding these desired values for the COM into the whole-body control framework. The estimated ZMP is different from the desired ZMP, as the LIPM model is a simplified version and because there are motions other than COM. Therefore, the reference ZMP is updated by the amount of difference between the desired ZMP and the estimated ZMP. The desired ZMP and the estimated ZMP converge to the same value by iteration, from which the desired position and velocity of the COM are computed. This process can be shown via the pseudo code below:

while $|r_{x,d} - r_{x,est}| \geq threshold$ **do**
  Calculate trajectory x_d, v_d using (16).
  Calculate desired torque using (7) with x_d, v_d.
  Calculate contact forces f_c and moments using (9).
  Calculate $r_{x,est}$ using (14).
  Update reference ZMP, $r_{x,ref} = r_{x,ref} + r_{x,d} - r_{x,est}$.
end while

The iteration result of the ZMP is shown in Fig. 2. The estimated ZMP converges to the desired value when the iteration number increases.

C. Planned Velocity Tracking Algorithm

The velocity of the COM becomes faster or slower depending on the modification of the COM trajectory introduced in the previous section. However, it is possible to increase or decrease the commanding velocity whenever the ZMP has a feasible region to change. This possibility occurs when the expected ZMP is within the supporting polygon, as illustrated in Fig.3.

If the desired velocity is modified while walking is taking place, the velocity of the robot on average will differ from the planned value. Therefore, our goal is to match the modified desired velocity on average with the planned velocity. This is accomplished by the following procedure. First, the difference between the modified desired velocity and the planned velocity is integrated over time. If the integrated value is negative and there are lower margins for the ZMP, the desired ZMP is set to the necessary value to compensate for the integrated value and thus increase the velocity of the COM until the integrated value becomes zero. Alternatively,

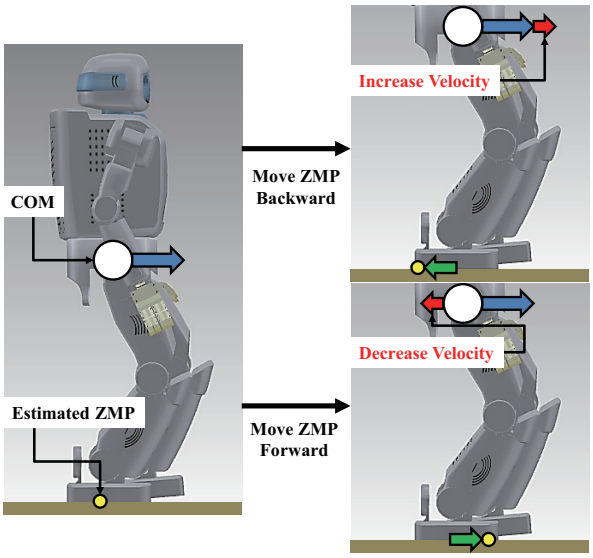

Fig. 3. **Illustration of the effect of changing ZMP**. When the ZMP moves backward during walking, the velocity of the COM become faster, and when the ZMP moves forward, the velocity of the COM become slower.

if the integrated value is positive, the desired ZMP value is set to value within the boundary of the ZMP to decrease the velocity.

The experimental result shown in Fig. 4 illustrates how the algorithm works. Fig.4 is the part of the walking with planned velocities of 0.063 [m/sec] and 0.156 [m/sec]. The planned velocity changes here at 0.8 [sec]. At 0.8 [sec], to follow the changed planned velocity, the desired ZMP takes on a lower value of the designed boundary, as shown in Fig. 4 (a), so that the COM is accelerated. The desired ZMP remains at the lower boundary value until the integration of the difference between the desired velocity and planned velocity became zero. After 0.88 [sec], the desired velocity is constant and the integration of the velocity difference becomes zero. In this region, the torque for maintaining a constant velocity is generated with the ZMP, which is inside of the boundary.

D. Lateral Motion

Lateral motion during walking is different from sagittal motion because the COM motion is supposed to oscillate between two feet. For this reason, the trajectory of lateral motion is computed using the LIPM model with a zero reference ZMP, unlike the sagittal motion. The equations of motion in the lateral direction is

$$\begin{bmatrix} y_d \\ T_c w_d \end{bmatrix} = \begin{bmatrix} \cosh(\frac{\Delta t}{T_c}) & \sinh(\frac{\Delta t}{T_c}) \\ \sinh(\frac{\Delta t}{T_c}) & \cosh(\frac{\Delta t}{T_c}) \end{bmatrix} \begin{bmatrix} y \\ T_c w \end{bmatrix}, \qquad (17)$$

where y_d and w_d are respectively the desired COM position and velocity for the lateral direction and the variables y and w are likewise the current COM position and velocity for the lateral direction.

One important aspect pertaining to lateral motion is determination of the switching time of the ZMP from one foot to the other. For simplicity, it is assumed to change instantly

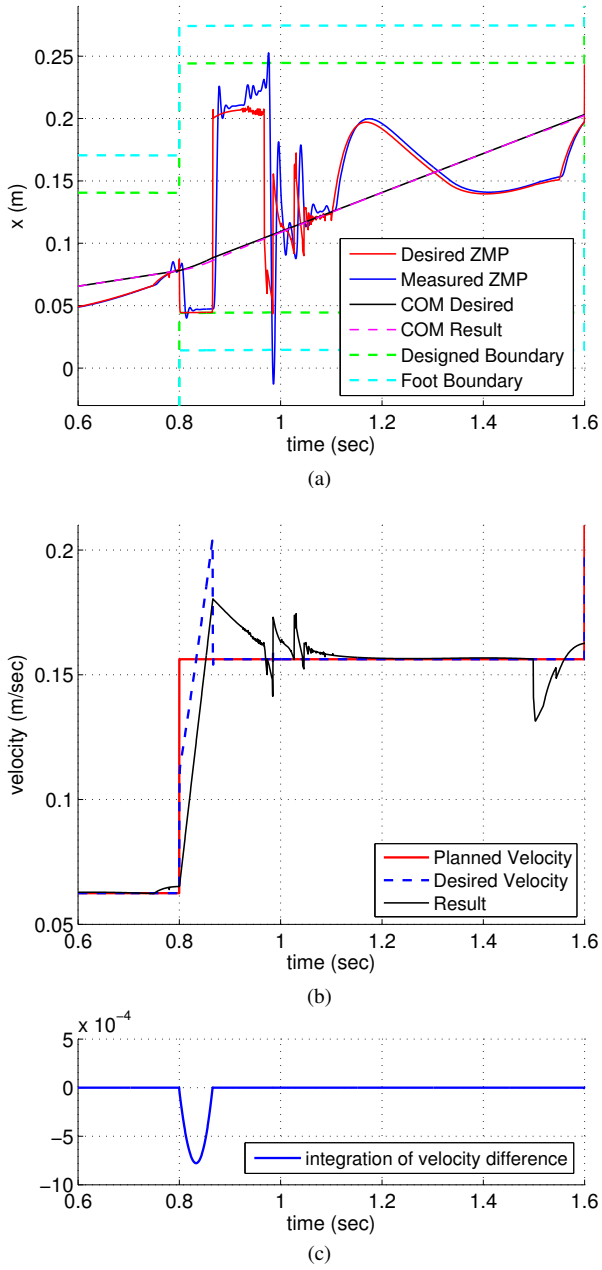

Fig. 4. **Walking simulation result for sagittal direction**. (a) COM and ZMP position, (b) COM velocity, and (c) integration of the difference between the desired velocity and the planned velocity.

at the switching time during the double support phase. The switching time is determined such that the velocity in the lateral direction is zero at the half-way point of the next single support duration. At every servo cycle at the double support phase, the velocity is checked as to whether it would be zero in the middle of the next single support phase if the ZMP is switched at the next servo cycle. If yes, the ZMP is switched at the next servo cycle. This can be computed using the following equation derived from (17).

$$w(T) = (\sinh(\frac{T}{T_c})y + \cosh(\frac{T}{T_c})T_c w)/T_c, \qquad (18)$$

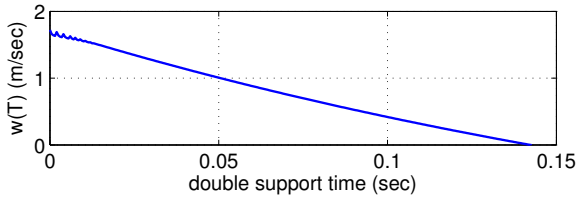

Fig. 5. **Expected velocity of lateral direction**. In this case, $w(T)$ becomes zero at 0.142 [sec] of the double support phase. This indicates that the ZMP switches from one foot to the other at that time.

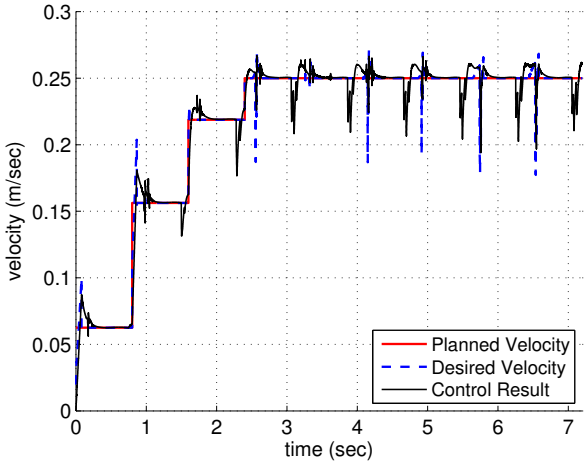

Fig. 6. **Trajectory of COM velocity for sagittal direction**.

where T is the duration from the next servo time to half of the single support time. This value is checked until it becomes zero, i.e., $w(T) = 0$. The plot in Fig. 5. shows the computed values of $w(T)$ during a double support time in our experiment.

IV. SIMULATION RESULTS

The proposed method is implemented in a simulation using the physics-based simulation software *RoboticsLab* [16]. The simulated humanoid robot model is MAHRU [17] which was developed by the Korean Institute of Science and Technology (Fig. 1). The robot has 25 joints. Each leg and arm of the robot has 6 DOF and the waist has 1 DOF (yaw). The length and width of the foot are 0.25 m and 0.14 m, respectively.

The whole-body control framework is used to control the COM position of the robot and the orientation of the trunk during the double support time. Additionally, the position and orientation of the swinging foot is controlled during the single support phases. The orientation of the trunk and foot was commanded to maintain their initial values at the onset of walking. The foot position trajectory was generated with a cubic spline function for each direction. The proposed algorithm is applied to walking while increasing the planned velocity in the sagittal direction. As shown in Fig. 6, the walking started in a stationary state and the walking speed for the sagittal direction was changed four times with step functions. The planned velocity was 0.063 [m/sec] when the

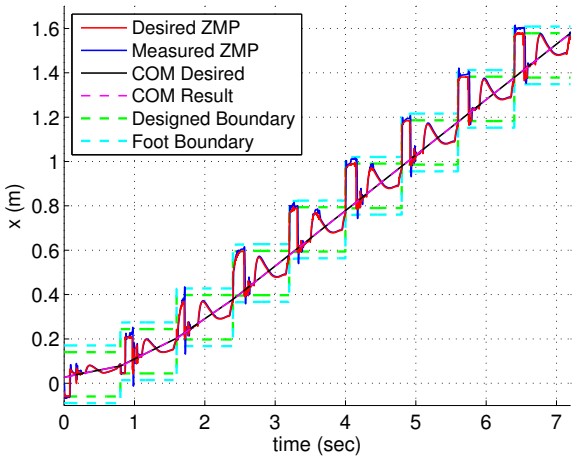

Fig. 7. **COM and ZMP for sagittal direction**.

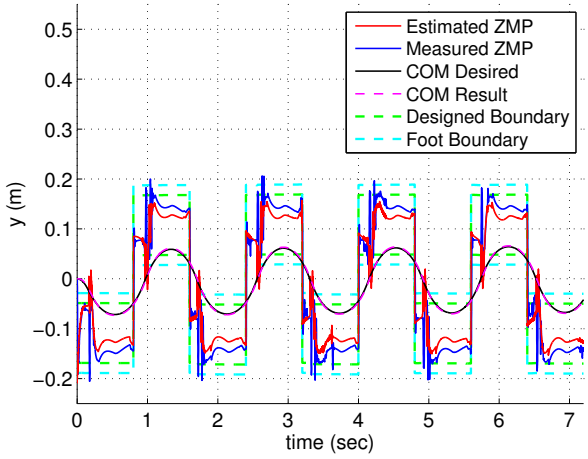

Fig. 8. **COM and ZMP for lateral direction**.

walking started, and it was changed to 0.156, 0.219, and 0.25 [m/sec] at 0.8, 1.6, and 2.4 [sec], respectively. The generated foot step lengths for the sagittal direction were set to 0.1, 0.15, and 0.2m for the first three steps. Then, the step length was repeated with the length set of 0.25m from the fourth step. The lateral motion was designed to repeat symmetric foot steps to maintain a relative distance of 0.22m from the other foot. The double and single support times were set to 0.16 [sec] and 0.64 [sec], respectively, for every walking period. The threshold for ZMP control with iteration in Section III-B was 0.001.

The experimental results using the proposed algorithm are plotted in Figs. 6 - 8. Fig. 6 shows the planned velocity and generated trajectory. The generated trajectory tracks the planned velocity well except for two situations. One is when the planned velocity changes and the other is when the contact state changes. The first situation is mainly due to the fact that the changes in the planned velocity are discontinuous. The second is because discontinuous contact

forces appear and vibration occurs at the moment the contact state changes suddenly. This occurred periodically after 0.16 [sec] with an interval of 0.8 [sec] and appeared periodically with a period of 0.8 [sec] after 1.5 [sec] as well. These events are caused by the collision between the swinging foot and the ground.

For the sagittal direction, the desired ZMP during walking went to the lower value of the designed boundary for the COM to accelerate, after the planned velocity changed. Then, it went inside of the foot after acceleration ended. This can be seen at the time between 0∼2.4 [sec] in Fig. 6 and Fig. 7. At 2.4 [sec], the planned velocity changed but the desired ZMP did not go to the lower boundary because the desired acceleration was small enough to be generated with the ZMP inside of the foot. After 2.4 [sec], the desired velocity became relatively large and effect of other motions on the ZMP also became too large to follow the velocity. Therefore the ZMP went out of the upper boundary at 4.08∼4.15 [sec], 4.87∼4.91 [sec], 5.64∼5.75 [sec], and 6.44∼6.54 [sec]. In this case, the desired ZMP was set to the designed upper boundary and the COM trajectory was modified to follow the desired ZMP. After that, during 4.15∼4.18 [sec], 4.92∼4.95 [sec], 5.75∼5.79 [sec], and 6.54∼6.58 [sec], the desired ZMP went to the lower boundary to reduce the integration of the difference between the modified desired velocity and the planned velocity to zero. During nominal walking, the planned velocity can be achieved with the available ZMP changes. This approach, however, may not generate the exact desired velocity when there is large whole-body motion or a disturbance. In such cases, it is desirable to change the desired velocity and the foot trajectories accordingly.

For lateral motion, the COM trajectory was generated by the equation used in Section III-D and the result of the execution is plotted in Fig. 8.

V. CONCLUSION

A forward walking strategy with a given planned velocity of the COM is proposed. The ZMP is estimated using whole-body dynamics and used to check the stability condition. When the estimated ZMP does not satisfy the stability condition, the COM trajectory is modified to change the estimated ZMP for stable walking. The proposed algorithm was implemented using a 25-DOF humanoid model in the physics-based simulation. The results is demonstrated that this strategy can generate walking with a changing velocity without the explicit planning of the COM trajectory.

The proposed approach is expected to be especially useful when there are whole-body motions such as motions of the arms. Our future work will include further experiments in such situations, synchronization of the COM velocity and foot step planning, and experimental validation with a physical robot.

REFERENCES

[1] M. Vukobratović, and J. Stepanenko, On the Stability of Anthropomorphic Systems, in Mathematical Biosciences, vol. 15, pp. 1-37, 1972.

[2] M. Vukobratović and B. Borovac, Zero-Moment Point - Thirty Five Years of Its Life. in Int. J. Humanoid Robotics, vol. 1, no. 1, pp. 157-173, 2004.

[3] J. Yamaguchi, E. Soga, S, Inoue, and A. Takanishi, Development of a Bipedal Humanoid Robot - Control Method of Whole Body Cooperative Dynamic Biped Walking -, in Proc. IEEE Int. Conf. Robotics and Automation, 1999, pp. 368-374.

[4] K. Hirai, M. Hirose, Y. Haikawa, and Toru Takenaka, The Development of Honda Humanoid Robot, in Proc. IEEE Int. Conf. Robotics and Automation, 1998, pp. 1321-1326.

[5] 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. Robotics and Automation, 2002, 31-37.

[6] 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. Robotics and Automation, 2003, pp. 1620-1626.

[7] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa, Biped Walking Pattern Generation allowing Auxiliary ZMP Control, in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2006, pp. 2993-2999.

[8] K. Nishiwaki, and S. Kagami, Simultaneous Planning of CoM and ZMP based on the Preview Control Method for Online Walking Control, in Proc. IEEE-RAS Int. Conf. Humanoid Robots, 2011, pp. 745-751.

[9] 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, in IEEE Trans. Robotics, vol. 24, no. 4, pp. 917-924, 2008.

[10] K. Erbatur, and O. Kurt, Natural ZMP trajectories for biped robot reference generation, in IEEE Trans. Ind. Electron, vol. 56, no. 3, pp. 835-845, 2009.

[11] B. Ugurlu, and A. Kawamura, Bipedal Trajectory Generation Based on Combining Inertial Forces and Intrinsic Angular Momentum Rate Changes: Eulerian ZMP Resolution, in IEEE Trans. Robotics, vol. 28, no. 6, pp. 1406-1415, 2012.

[12] J.-H. Kim, and J. -H. Oh, Walking Control of the Humanoid Platform KHR-1 based on Torque Feedback Control, in Proc. IEEE Int. Conf. Robotics and Automation, 2004, pp.623-628.

[13] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking, in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2010, pp. 4489-4496.

[14] K. Miura, M. Morisawa, F. Kanehiro, S. Kajita, and K. Yokoi, Human-like Walking with Toe Supporting for Humanoids, in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 2011, pp. 4428-4435.

[15] J. Park, and O. Khatib, Contact Consistent Control Framework for Humanoid Robots, in Proc. IEEE Int. Conf. Robotics and Automation, 2006, pp. 1963-1969.

[16] ROBOTICSLAB, http://www.rlab.co.kr/.

[17] B. You, D. Kim, C. Kim, Y. Oh, M. Jeong, and S. Oh. Network-based humanoid mahru as ubiquitous robotic companion, in Proc. Int. Fed. Automatic Control, 2008, pp. 724-729.