2010 IEEE-RAS International Conference on Humanoid Robots
Nashville, TN, USA, December 6-8, 2010

# Convolution-Sum-Based Generation of Walking Patterns for Uneven Terrains

H. Andy Park, Muhammad A. Ali, and C. S. George Lee
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-2035
{andypark,ali2,csglee}@purdue.edu

*Abstract*— **In generating walking patterns for humanoid robots, a Center-of-Mass trajectory is usually derived from the desired Zero-Moment-Point (ZMP) trajectory. One way to accomplish this is the use of the preview-control method, which tracks the desired ZMP trajectory while minimizing the jerk. Another method, which is more computationally efficient, is based on the convolution-sum method. Although this method is simple to implement, the resulting motion could be jerky. In this paper, we utilize the convolution-sum method to generate walking patterns for slopes and stairs walking while minimizing jerky motions. Furthermore, we extend the method to generate walking patterns for non-uniform terrain walking. This is accomplished by defining certain coordinate frames and maintaining the right-foot posture necessary for achieving robust walking. Computer simulations utilizing Webots were performed to validate the proposed convolution-sum method for the generation of walking patterns for a HOAP-2 humanoid robot.**

## I. Introduction

To realize walking in uneven terrains, it is obvious that humanoid robots should be equipped with two important mechanisms. One is to get sufficiently accurate information of the surrounding environment and terrain in real time either through visual input or some other sensors. The other one is to plan and execute robust walking according to the acquired environment and terrain information [1], [2], [3]. Although many walking patterns have been proposed for humanoid robots to walk on unknown terrains [4], [5], [6], [7], most of these techniques are applicable to slightly uneven terrains as indicated in [6]. To move around on uneven ground with large variations like stairs or steep slopes, real-time terrain information is essential. We assume that real-time terrain information is available through necessary sensory inputs.

In this paper we deal with the problem of walking on uneven terrains with large variations. For generating walking patterns based on ZMP, a desired ZMP trajectory is first designed for a given placement of footsteps, then a Center-of-Mass (CoM) trajectory is derived such that the resulting ZMP would follow the desired or reference ZMP trajectory [8], [9]. One way to do this is by using the preview-control method introduced by Kajita et al. [10], which was used by Huang [11] to generate walking patterns for walking on slopes and stairs. Although the preview-control method tracks the desired ZMP trajectory reasonably well, it needs to solve Ricatti's equation for the optimal control problem. Wieber et al [12] proposed a method, which solves the optimal control problem through simpler matrix manipulations instead of solving the more complex algebraic Ricatti's equation.

Recently, another method, introduced by Kim [13], efficiently solves the problem of finding the CoM trajectory. It uses a convolution-sum-based algorithm to derive the CoM trajectory in real time from a given ZMP trajectory. This convolution-sum-based algorithm is able to track the ZMP trajectory exactly in computer simulations. Unfortunately, the generated CoM trajectory from the convolution-sum method has jerky acceleration of CoM in certain conditions [13]. In this paper, we extend the convolution-sum-based method to generate walking patterns for walking on level ground and on uneven terrains like slopes and stairs. We also minimize the jerky acceleration of CoM to accomplish smooth walking.

For the remaining of the paper, we first discuss the concept and analysis of the convolution-sum method as compared to the preview-control method. In Section III, we propose a flexible walking algorithm based on the proposed convolution-sum-based method to generate walking patterns for known uneven terrains such as slopes and stairs utilizing the scheme of coordinate frames. In Section IV, extensive computer simulations of walking patterns for a HOAP-2 humanoid robot were performed to validate the proposed walking-pattern generation on uneven terrains. Finally, Section V summarizes the conclusions of the proposed work.

## II. Convolution-Sum Method

The convolution-sum method [13] is based on a linear inverted-pendulum model with a point mass as shown in Fig. 1. Let us first restate the Zero-Moment-Point (ZMP) equations, which describe the relation between the ZMP and the Center-of-Mass (CoM) position and acceleration. It assumes that the height ($H$) of CoM remains constant in the vertical direction. Using this assumption, the $x$ and $y$ coordinates of ZMP can be described by the following equations [10], [13], [14]:

[†]This work was supported in part by the National Science Foundation under Grant IIS-0916807. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

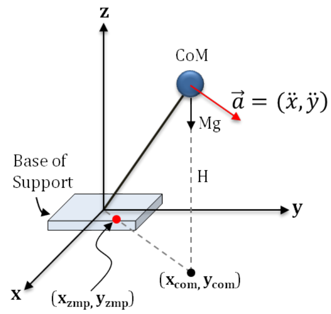


Fig. 1. A linear inverted-pendulum model.

$$x_{zmp} = x_{com} - \frac{H}{|\mathbf{g}|}\ddot{x}_{com} \tag{1}$$

$$y_{zmp} = y_{com} - \frac{H}{|\mathbf{g}|}\ddot{y}_{com} \tag{2}$$

where $|\mathbf{g}|$ is the magnitude of the acceleration due to the gravity $\mathbf{g}$, and ($x_{zmp}$, $y_{zmp}$) is the position of ZMP trajectory. Throughout this paper, unless specified otherwise, $x$-axis is in the direction of forward motion of the humanoid robot, $y$-axis is along the lateral motion, and $z$-axis is vertically upwards away from the supporting ground.

### *A. Finding CoM trajectory from ZMP reference Trajectory using Convolution Sum*

The above Eqs. (1) and (2) show that if the CoM trajectory is known, then the ZMP trajectory can be easily computed. However, for the generation of walking patterns, we need to solve the inverse problem; that is, we are interested in deriving the CoM trajectory from the desired ZMP trajectory. Since Eqs. (1) and (2) are second-order differential equations in $x_{com}$ and $y_{com}$, respectively, we can treat the ZMP trajectory as an input to a system and the CoM trajectory as the output or response of the system. The output of a linear time-invariant system is actually the convolution of the input signal with its impulse response. So the idea here is if we can find the impulse response of this system, we can convolve any given ZMP trajectory with this impulse response to obtain the CoM trajectory. Thus, an acausal step response, which satisfies the above equations, is first obtained, then from the step response, a discrete-time impulse response, $h[n]$, is obtained [13],

$$h[n] = h[0] \cdot e^{-|n|\lambda T}, \tag{3}$$

where $h[0] = \frac{1}{2}(e^{-\lambda T} - 1)e^{-\frac{\lambda T}{2}} = \sinh(\frac{\lambda T}{2})$, $\lambda = \frac{|\mathbf{g}|}{H}$, and $T$ is the sampling period.

Using the above impulse response truncated at a sufficiently large $N$, the CoM trajectory can be found from the reference ZMP trajectory by using the convolution-sum method,

$$x_{com}[n] = \sum_{k=-N}^{k=N} x_{zmp}[n-k]h[k]. \tag{4}$$

If $N$ is chosen to be sufficiently large so that the effect of truncation of impulse response is minimal, the convolution-sum method in principle tracks the ZMP trajectory exactly. Kim [13] has also derived a recursive formula for computing the above convolution-sum method, which is computationally more efficient for the online generation of CoM trajectory.

### *B. ZMP Preview-Control Method*

The ZMP preview-control scheme [10], which generates dynamically stable motions for a humanoid robot, is also based on a linear inverted-pendulum model for designing an optimal servo controller. In this method, a new variable is introduced, which is the derivative of CoM acceleration. If the CoM trajectory has abrupt changes in acceleration, then the resulting motion will be jerky. So the derivative of CoM acceleration quantifies the jerkiness of motion [12]. Hence, the strategy for the generation of walking motion in the preview-control method is to track the ZMP trajectory closely while minimizing the derivative of CoM acceleration. In general, these two goals are competing objectives; that is, minimizing errors in the ZMP trajectory may result in large jerkiness, and minimizing the jerkiness may result in large errors in the ZMP trajectory. Hence, the control problem is formulated as the linear quadratic regulator (LQR) problem.

$$\min_{\dddot{x}^{(k)}_{com}, \dddot{x}^{(k+1)}_{com}, \ldots} \sum_{i=k}^{\infty} \frac{1}{2}Q(x^{(i)}_{zmp} - x^{(i)\,ref}_{zmp})^2 + \frac{1}{2}R\dddot{x}^{(i)2}_{com} \tag{5}$$

where the parameters $Q$ and $R$ are appropriate weights, which reflect the relative importance of keeping the error in ZMP trajectory small versus keeping the jerkiness to a minimum. For this method we have to solve the Ricatti's equation to obtain a solution to the above problem.

### *C. Problems with Convolution-Sum Method*

Although the convolution-sum method tracks the ZMP trajectory exactly, its CoM trajectory may have large jerky motions. As shown in Fig. 2, when the ZMP trajectory is a step input, the resulting CoM trajectory has an abrupt change in the CoM acceleration. This means that the derivative of CoM acceleration is quite large, which results in large jerky motions. Thus, one needs to keep the jerk, or the derivative of CoM acceleration, within acceptable bounds.

We found that instead of using a ZMP trajectory, which has a step change, we use a ZMP trajectory that rises somewhat gradually, the resulting third derivative of CoM trajectory gets reduced significantly as shown in Fig. 3. This shows that the jerk has some dependency on the shape of desired ZMP trajectory. This becomes evident when we take the time derivative of Eq. (1) and obtain

$$\dot{x}^{ref}_{zmp} = \dot{x}_{com} - \frac{H}{|\mathbf{g}|}\dddot{x}_{com} \tag{6}$$

Rearranging Eq. (6), we get

$$\dddot{x}_{com} = \frac{|\mathbf{g}|}{H}(\dot{x}_{com} - \dot{x}^{ref}_{zmp}) \tag{7}$$

Equation (7) shows that the third derivative of $x_{com}$ (i.e., $\dddot{x}_{com}$) is a function of CoM velocity $\dot{x}_{com}$ and the first derivative of ZMP trajectory $\dot{x}_{zmp}$. Since the CoM velocity

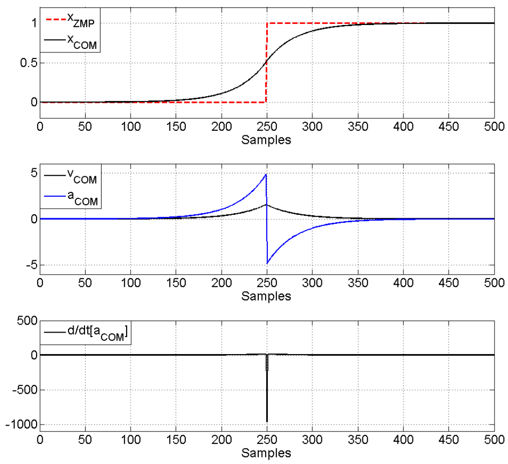



Fig. 2. Step reference ZMP trajectory with Ts = 10 ms and $l$ = 1 m, where $\lambda = \sqrt{|\mathbf{g}|/l} = 3.1315$

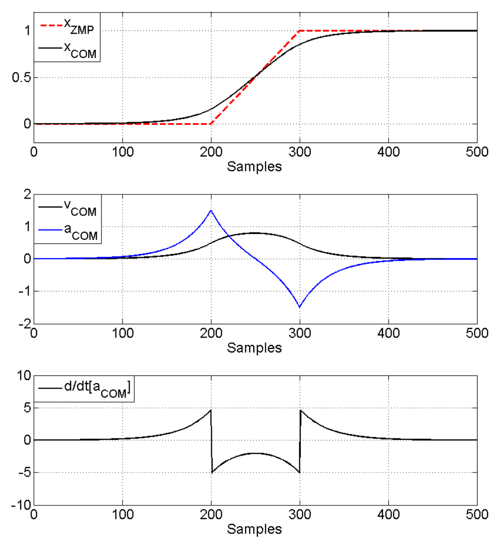



Fig. 3. Modified reference ZMP trajectory with Ts = 10 ms and $l$ = 1 m, where $\lambda = \sqrt{|\mathbf{g}|/l} = 3.1315$

always remains within a limited range, and if $\dot{x}_{zmp}$ is large, then the jerk can be large. Hence, given a bound on the value of jerk (i.e., third derivative of $x_{com}$ or $y_{com}$), we can find a bound on the first derivative of $x_{zmp}$ and $y_{zmp}$. From Eq. (7), we find that the maximum absolute value of $\dddot{x}_{com}$ at any instant of time can be at most,

$$\max|\dddot{x}_{com}| = \frac{|\mathbf{g}|}{H}(|\dot{x}_{com}| + |\dot{x}_{zmp}|) \tag{8}$$

Hence, the upper bound on $x_{zmp}$ and $y_{zmp}$ velocities can be written as

$$|\dot{x}_{zmp}| \leq \frac{H}{|\mathbf{g}|}\max|\dddot{x}_{com}| - \max|\dot{x}_{com}| \tag{9}$$

$$|\dot{y}_{zmp}| \leq \frac{H}{|\mathbf{g}|}\max|\dddot{y}_{com}| - \max|\dot{y}_{com}| \tag{10}$$

where $\max|\dot{x}_{com}|$ and $\max|\dot{y}_{com}|$ are maximum attainable CoM velocity during the walking cycle in $x$ and $y$ direction, respectively.

This implies that if we design the desired ZMP trajectory such that the ZMP velocity is bounded by some value, then we shall have jerkiness in motion within allowable limits. Let us consider a single-support phase walking scenario with footsteps as shown in Fig. 4(a). One way is to design the ZMP trajectory such that the ZMP lies at the center of supporting foot, and it moves to the other foot instantly when the supporting foot changes. The resulting ZMP trajectory profile will look like as in Fig. 4(b). We can see that the ZMP trajectory has step-size changes, which are not desirable since they will result in large jerky motions.

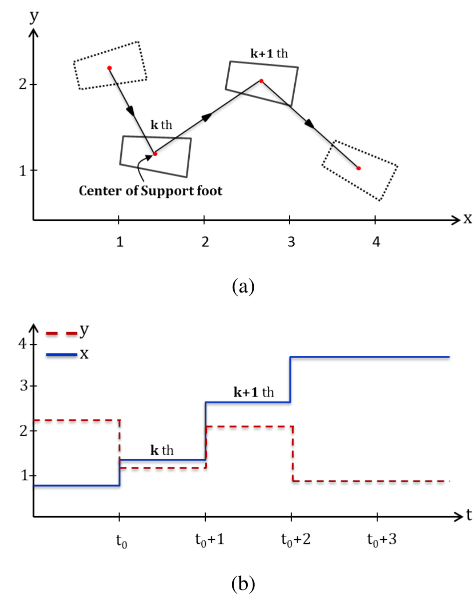



Fig. 4. (a) Footsteps, (b) $X_{zmp}$ profile and $Y_{zmp}$ profile.

If we introduce a double-support phase (DSP) between two single-support phases (SSP) and appropriately choose the duration of DSP, we can design the ZMP trajectory to have a bounded ZMP velocity. It should be noted that the ZMP velocity is very low during a SSP and is only high when the ZMP has to move from one supporting foot to the other during a DSP. This will result in balanced motions no matter how slow the ZMP moves from the previous foot to the next foot because we shall ensure that this ZMP transition takes place during the DSP only. This design of the ZMP trajectory will lie within the support convex hull bounded by the two feet. The duration of the DSP of $k$th walking step can be found as follows.

$$t_{DSP} \geq \frac{x_{supp}^{(k+1)} - x_{supp}^{(k)}}{\max|\dot{x}_{zmp}|} \tag{11}$$

where $x_{supp}^{(k)}$ is the $x$ coordinate of supporting-foot position in $k$th walking step. Also we should have

$$t_{DSP} \geq \frac{y_{supp}^{(k+1)} - y_{supp}^{(k)}}{\max|\dot{y}_{zmp}|}. \tag{12}$$

The greater of the above two values is selected as $t_{DSP}$. Calculating the duration of a DSP ensures that the ZMP velocity remains within the upper bound defined by Eqs. (9) and (10).

For the HOAP-2 robot that we used in our computer simulations, the parameter values are $H = 0.313$m, $\max|\dot{x}_{com}| = 0.2$m/s, and $|g| = 9.8066$m/s$^2$. For example, if we choose the upper bound on $|\dddot{x}_{com}| = 20$m/s$^3$ and for the first step, $x_{supp}^{(0)} = 0$ and $x_{supp}^{(1)} = 0.08$m, then using the above equations we get $\max|\dot{x}_{zmp}| = 0.438$ m/s, which in turn gives the lower bound of the duration of DSP as $t_{DSP} = 183$ msec.

## III. Convolution-Sum-Based Flexible Walking Algorithm for Known Terrains

Whether it is the preview-control method or the convolution-sum method for generating walking patterns, the trajectories generated must be referenced to some coordinate frame. One can use the world coordinate frame as the reference coordinate frame. However, by doing this way, it will become more complicated to plan the trajectories. This is mainly because if the robot has been walking for a while, we need to maintain the complete history of walking in order to determine its current location with reference to the world coordinate frame (**W**). We need to do it because in this method everything, including foot positions, trajectories of CoM and ZMP and so on, would need to be referenced to the world coordinate frame. Thus, it is better and more efficient to use some local coordinate frame as a reference for planning trajectories.

Since there exist many choices of local coordinate frames, one can assign the reference coordinate frame to the center of the trunk of the robot. But this is also not a good choice for planning trajectories, especially if it is omni-directional walking. Furthermore, if performing a turning motion, the reference coordinate frame itself will be changing its orientation as well. Hence, the best choice is to choose a local coordinate frame that will remain stationary with respect to the ground during a phase of the walking cycle. A walking cycle has two distinct phases — a single support phase (SSP) and a double support phase (DSP). During a SSP, the robot is only supported by one leg and the other leg is in a swing mode. If we assume that there is sufficient friction between the support foot and the ground (i.e., there is no slippage), then the support foot can be assumed to be stationary with respect to the ground during a SSP. Hence, we can assign the reference coordinate frame to the support foot as shown in Fig. 5 and calculate all the trajectories belonging to this walking cycle with respect to this coordinate frame. We shall denote this coordinate frame as a trajectory coordinate frame or just the **T** frame. Here we should note that the support foot changes from left to right and vice versa once in every walking cycle. So does the **T** coordinate frame.

Strom et al [15] proposed a similar scheme for omni-directional walking on a level ground. As we shall show, selecting the reference coordinate frame at the support foot will make it easier to implement walking on uneven ground as well as walking on slopes and stairs. For walking on slopes and stairs, Huang et al [11] have proposed a method in which they have to change the ZMP model whenever the robot walks on a slope. The change proposed in the ZMP

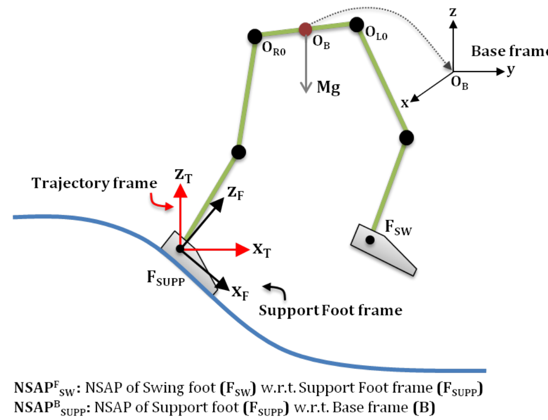



Fig. 5. Coordinate frames being used.

model is convenient in the sense that it is a simple change and the computation of trajectories for slopes remains very similar to that for the level ground. But the ZMP model has to be changed whenever the robot encounters a different slope and connecting two trajectories over slope discontinuities is rather cumbersome. In our approach of selecting the reference coordinate frame at the support foot, we do not have to change the ZMP model for changing slopes as well as to all sorts of terrains like stairs and uneven ground.

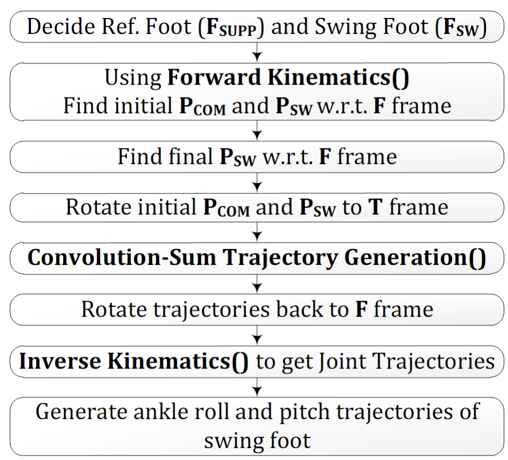



Fig. 6. Chart of the proposed convolution-sum walking algorithm.

The proposed convolution-sum walking algorithm of generating walking patterns is shown in Fig. 6. The proposed algorithm assumes that the terrain information is available in real time and the position of the robot relative to the terrain is also known at all instants of time with the help of some sensory information system. For a given initial configuration of the robot's feet and posture, the foot-step planner decides the next foothold based on the terrain information. Here we do not discuss the operation of the footstep planner, but just assume that the next planned foot step is available. It should be noted that the next planned footstep needs not be on the same level, it can be at a lower or higher level depending upon the terrain. Based on the next footstep, the support foot and the swing foot are determined. A new coordinate frame, which we mentioned as the trajectory coordinate frame or the **T** frame, is attached to the support foot such that the origin of the trajectory coordinate frame coincides with the origin of the support-foot coordinate frame ($\mathbf{F}$), but its $z$-axis is always parallel to the gravity. That is why using this trajectory coordinate frame, we can generate trajectories for both level ground as well as slopes and stairs because the coordinate frame always remains aligned with the gravity, and we do not have to change the ZMP model as in [11]. Using the forward kinematics, the initial position of CoM and the initial position of the swing foot are determined with reference to the support-foot coordinate frame. Note that the support-foot coordinate frame ($\mathbf{F}$) is not the same as the trajectory coordinate frame ($\mathbf{T}$). Although their origins coincide, the coordinate frame $\mathbf{T}$'s $z$-axis is always aligned with the gravity and the orientation of the $\mathbf{F}$ coordinate frame changes with the foot orientation. Using the rotation matrix ${}^{T}\mathbf{R}_{F}$, the initial position of CoM and the initial position of the swing foot can be rotated to the $\mathbf{T}$ frame because from now on everything will be referred to the $\mathbf{T}$ frame. Then the ZMP trajectory is designed for the current step according to the details in Section II. After that the CoM trajectory is obtained using the convolution-sum method.

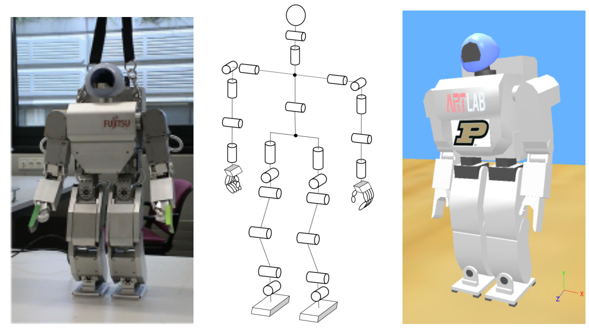
The swing-foot trajectory is calculated by a sinusoidal interpolation between the swing-foot initial position and the final position using a raised-cosine function for reducing the impact when the swing foot lands on the ground. Once the CoM trajectory and the swing-foot trajectory are obtained, joint trajectories are calculated using the inverse kinematics [16].

Finally, to ensure that when the swing foot lands on the ground, it is parallel to the ground, the available terrain information is used to calculate the gradient of the terrain surface at the landing site, and from that the required ankle joint angles (pitch and roll) of the swing foot are calculated. Then the pitch- and roll-angle trajectories are calculated by sinusoidally interpolating between the initial-angle values and the final-angle values required for safe landing.

## IV. Computer Simulations

Using the dynamic simulation software Webots, we evaluated the proposed convolution-sum walking algorithm on a HOAP-2 robot as shown in Fig. 7. HOAP-2 robot is 50 cm in height and weighs about 7.0 kg. It has 25 degrees of freedom (DoFs), 6 in each leg, 4 in each arm, 1 in each hand, 1 in the waist and 2 in the neck. The kinematic diagram of a HOAP-2 robot is shown in Fig. 7(b).

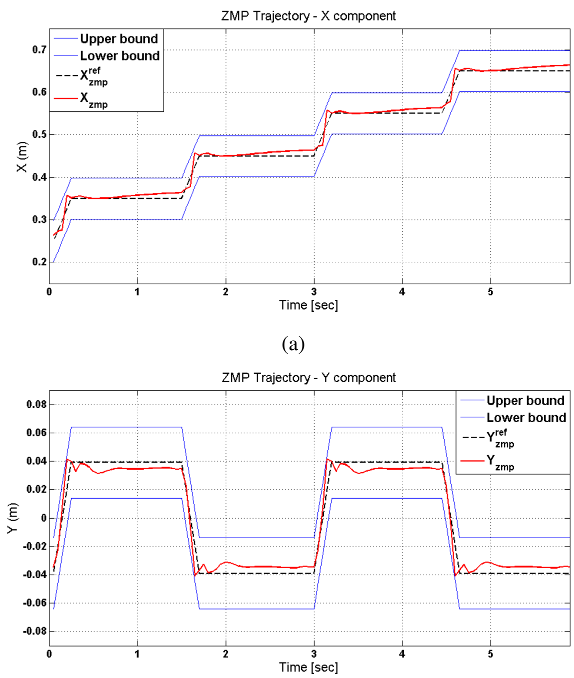
In our computer simulations, the real-time ZMP trajectory is measured by 4 force sensors placed under each foot of the robot. It is shown in Fig. 8 and the corresponding measured CoM trajectory and the resulting jerk is shown in Fig. 9. The step length is 0.1 m, the step duration is 1.5 sec, and the sampling period is 50msec. In the proposed convolution-sum trajectory generation function, we have $\lambda = 5.596$. The time constant of impulse response is $\tau = 1/\lambda = 0.179$. Therefore, the impulse response is truncated to $t = 5\tau$, which corresponds to having approximately $N = 30$ samples of impulse response in discrete time. The upper- and lower-bound indicate the stability margins in which ZMP can move



(a) (b) (c)

Fig. 7. (a) HOAP-2, (b) Kinematic diagram, and (c) HOAP-2 model in Webots.

without falling down, and they depend upon the size of the foot.



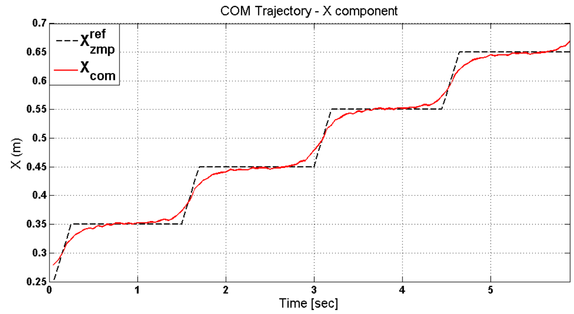

(a)

(b)

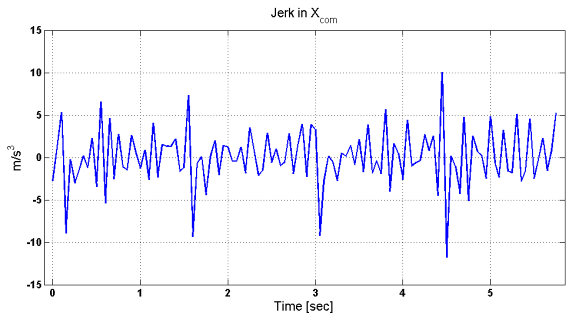
Fig. 8. Trajectory of ZMP.

As shown in Fig. 8, the actual ZMP trajectory measured by force sensors remains within the upper and lower bounds of stability margin. However, it is not exactly following the desired reference ZMP trajectory because of the difference in the dynamic model between the real robot and the linear inverted-pendulum model that we used. Since the DSP duration was determined in such a way that the jerk in the resulting motion remains within acceptable bounds, we can see from Fig. 9(b) that the jerk in the measured CoM motion is well within the bound of $\pm 20\text{m/s}^3$. Since the CoM trajectory was measured by using a position sensor attached to the CoM of the robot in the simulation software, we have some noise in the graphs of $x_{com}$ and the jerk of $x_{com}$.

In order to show the effectiveness of the proposed convolution-sum walking algorithm, we performed four simulations: Upslope/Downslope walking from a flat terrain, and upstair/downstair walking and transition walking from $5°$



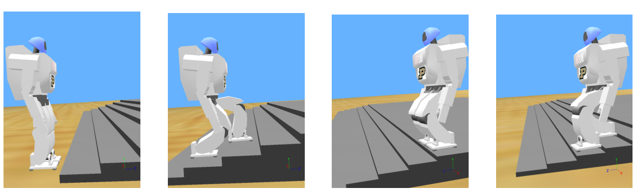

(a)




(b)

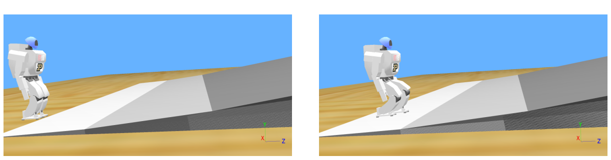
Fig. 9. Trajectory of CoM and jerk in X direction

to 10° to 15° slopes. The snapshots of these simulations are shown in Figs. 10-11. The simulations show that the proposed convolution-sum walking algorithm can be successfully applied for walking on stairs and grounds with changing slopes.

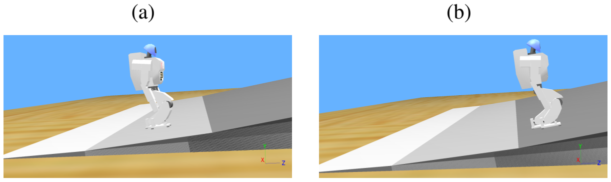


(a) (b) (c) (d)

Fig. 10. Snapshots of upstair walking (height=30mm, length=125mm) and Snapshots of downstair walking (height=15mm, length=100mm)



(a) (b)



(c) (d)

Fig. 11. Snapshots of transitional-upslope walking from 5° to 10° to 15°.

## V. Conclusions

In this paper, we extended and utilized the convolution-sum method to generate walking patterns for uneven terrains while maintaining the jerky motion of the generated CoM trajectory to within an acceptable bound. The proposed convolution-sum walking algorithm assumes that the unknown terrain information is available in real time. To improve the computational efficiency of the proposed walking algorithm, the proposed algorithm selects the support-foot coordinate frame as the reference coordinate frame, which remains stationary with respect to the ground during a phase of the walking cycle, for planning the trajectories. It has been demonstrated through computer simulations on Webots that it can enable successful walking on stairs and grounds with changing slopes.

## References


[1] J. Gutmann, M. Fukuchi, and M. Fujita, "Stair climbing for humanoid robots using stereo vision," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2004, pp. 1407–1413.

[2] A. Chilian and H. Hirschmuller, "Stereo camera based navigation of mobile robots on rough terrain," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2009, pp. 4571–4576.

[3] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2009, pp. 167–172.

[4] S. H. Hyon and G. Cheng, "Simultaneous adaptation to rough terrain and unknown external forces for biped humanoids," in *Proc. of the IEEE-RAS Intl. Conf. on Humanoid Robots*, November 2007, pp. 19–26.

[5] B. G. Son, "Impedance control for biped robot walking on uneven terrain," in *IEEE Intl. Conf. on Robotics and Biomimetics*, 2009, pp. 239–244.

[6] M. Ogino, H. Toyama, and M. Asada, "Stabilizing biped walking on rough terrain based on the compliance control," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2007, pp. 4047–4052.

[7] J. Kim, I. Park, and J. Oh, "Walking control algorithm of biped humanoid robot on uneven and inclined floor," *Journal of Intelligent and Robotic Systems*, vol. 48, no. 4, pp. 457–484, 2007.

[8] K. Nagasaka, H. Inoue, and M. Inaba, "Dynamic walking pattern generation for a humanoid robot based on optimal gradient method," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 6, 1999, pp. 908–913.

[9] J. Yamaguchi, S. Inoue, D. Nishino, and a. Takanishi, "Development of a bipedal humanoid robot having antagonistic driven joints and three DOF trunk," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 1998, pp. 96–101.

[10] 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 International Conference on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1620 – 1626.

[11] W. Huang, C.-M. Chew, Y. Zheng, and G.-S. Hong, "Pattern generation for bipedal walking on slopes and stairs," in *Proc. of the IEEE-RAS Intl. Conf. on Humanoid Robots*, 2008, pp. 205 –210.

[12] P. B. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," in *Proc. of the IEEE-RAS Intl. Conf. on Humanoid Robots*, Dec. 2006, pp. 137–142.

[13] J. H. Kim, "Walking pattern generation of a biped walking robot using convolution sum," in *Proc. of the IEEE-RAS Intl. Conf. on Humanoid Robots*, Nov. 2007, pp. 539 –544.

[14] N. Naksuk and C. S. G. Lee, "Utilization of Movement Prioritization for Whole-Body Humanoid Robot Trajectory Generation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, April 2005, pp. 1079–1084.

[15] J. Strom, G. Slavov, and E. Chown, "Omnidirectional walking using ZMP and preview control for the nao humanoid robot," in *RoboCup 2009: Robot Soccer World Cup XIII*. Springer, 2009, pp. 378–389.

[16] M. A. Ali, H. A. Park, and C. S. G. Lee, "Closed-Form Inverse Kinematic Joint Solution for Humanoid Robots," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.