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

A Hybrid Walk Controller for Resource-Constrained Humanoid Robots

Seung-Joon Yi*, Dennis Hong† and Daniel D. Lee*

Abstract— **Zero moment point (ZMP) preview controller is a widely adopted method for bipedal locomotion. However, for robots which are resource constrained or working in dynamic environments, simple reactive walk controllers are still favored as ZMP preview controllers have more control latency and are computationally more demanding. In this work, we present a hybrid walk controller that dynamically switches between a reactive walk controller based on the analytic solution of the linear inverted pendulum model and a ZMP preview controller that uses future foothold positions for more demanding tasks. The boundary conditions of center of mass (COM) state are considered in the optimization process of the ZMP preview controller to ensure a seamless transition between two controllers. We demonstrate the controller in a physically realistic simulations, as well as experimentally on a small humanoid robot platform.**

Keywords: humanoid robot, reactive walk controller, ZMP preview walk controller, push recovery

I. Introduction

The ultimate goal of humanoid robots is to work in human environment without special modification, and that is the very reason they are designed as humans. This requires a robust locomotion capability in unstructured environment, which may include walking over uneven surfaces, climbing stairs, stepping over obstacles or crossing gaps.

Human handles such hard terrains by first determining the future foothold positions and then dynamically plan its whole body movement using those foothold positions. This is the basis of the widely used zero moment point (ZMP) preview based walk controllers, which use the future foothold positions to generate an optimal center of mass (COM) trajectory that keeps the robot dynamically stable during the gait. This method has been implemented on various humanoid robot platforms and has been successfully used for various tasks such as stepping over obstacles [1] and stair climbing [2]. However, the main drawback of this method is that as it requires future ZMP trajectory, it typically has a few steps of control latency which does not allow for reactive movement needed for push recovery or dynamic environments. Also this method relies on an optimization over the future time steps, which generally requires more computational power than simple methods.

On the other way, when the terrain is not demanding, human uses a simple, reactive walking without considering future foothold positions. A number of bipedal walk controllers are based on such a reactive stepping. One variant of these approaches is the central pattern generator (CPG) based approach, which uses a number of simple basis functions to generate the COM trajectory without explicitly considering the ZMP criterion. Due to its simplicity, this approach has been widely used for small, resource constrained humanoid robots [3], [4], and also on the the Hubo full sized humanoid robot with help of feedback stabilization [5]. Another variant is the analytical ZMP approaches, which use the analytical solution of the ZMP equation to generate COM trajectory for each step, assuming a ZMP trajectory represented by limited order polynomials [6], [7], [8], [10]. Both approaches have little control latency and are computationally efficient, but they are less stable than ZMP preview method as they can have ZMP fluctuation with the change of walk velocity.

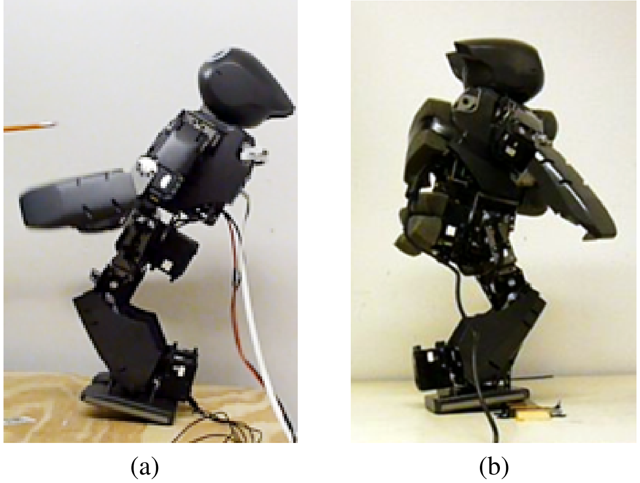


Fig. 1. Two different requirements for bipedal walk controller. (a) It should be able to take reactive step to handle external disturbance. (b) It should be able to generate a dynamically stable series of gait utilizing future foothold positions.

In this work, we suggest a hybrid walk controller that can switch between a reactive walk controller that can instantaneously change the walk velocity and a preview based walk controller that can handle a harder terrain using future foothold information. To generate a continuous COM trajectory over the transition, we extend the performance index of ZMP preview controller so that it considers the boundary conditions of the reactive controller. We validate our approach in a physically realistic simulation using a simulation model of the THOR full sized robot we are currently building for the DARPA Robotics Challenge (DRC), as well as experimentally on the DARwIn-OP small humanoid robot.

The remainder of the paper proceeds as follows. Section II describes the outline of the control architecture and gives a brief introduction of each components. Section III explains

* GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 {yiseung,ddlee}@seas.upenn.edu
† RoMeLa Laboratory, Virginia Tech, Blacksburg, VA 24061 dhong@vt.edu

978-1-4799-2619-0/13/$31.00 ©2013 IEEE

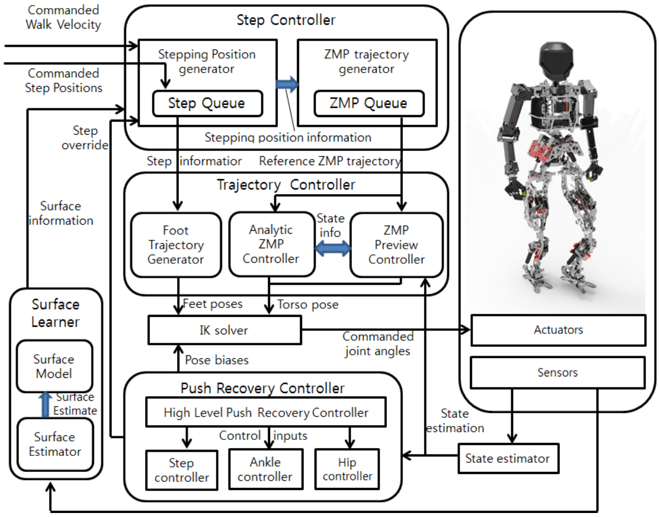


Fig. 2. The overview of the suggested motion controller. It has step controller, trajectory controller, push recovery controller and surface learner as submodules.

the step controller that generates stepping information and reference ZMP trajectory from control input. Section IV presents the detail of two COM trajectory methods we use. Section V describes how two COM trajectory generation methods can be switched during locomotion. Section VI and section VII shows results using a physics-based simulation and experiments using the DARwIn-OP small humanoid robot. Finally, we conclude with a discussion of potential future directions arising from this work.

II. Outline of the Control Architecture

Figure 2 shows the overview of the suggested control architecture, which is an extension of our previous control architecture that supports push recovery and uneven terrain walking [11]. It has four main components: the step controller that receives the user input to generate the step locations and ZMP trajectories, the trajectory controller that generates feet and torso trajectories, the push recovery controller that handles reactive balancing behaviors against external perturbation, and the surface learning controller that updates the surface model for uneven terrain based on the sensory information. In this paper, we mainly focus on the step and trajectory controllers, which now supports heterogeneous input signals and two different trajectory generation method. We will cover each controller in following sections.

III. Step Controller

Step controller has two main roles. It uses the control inputs to determine the foothold positions, and generates the ZMP trajectory that keeps the robot dynamically stable during the step. For our hybrid walk controller, it should be able to handle two heterogeneous control inputs: commanded walk velocity for reactive control and commanded step positions for preview control. We discuss each part in more detail in following subsections.

A. Reactive Step Generation

For reactive control, the step controller uses the walk velocity as the control input to generate the next foothold positions [6]. The i_{th} reactive step is defined the same way

$$STEP_i^R = \left\{SF_i, L_i, R_i, L_{1+1}, R_{i+1}, t_{STEP}^i\right\}, \tag{1}$$

where $SF_i \in \{LEFT, RIGHT\}$ denotes the support foot, L_i, R_i the pose of left and right feet in (x,y,θ), and t_{STEP} the duration of the step. To get a closed form COM trajectory, we confine the ZMP trajectory $p_i(\phi)$ to have the following piecewise linear form

$$p_i(\phi) = \begin{cases} C_i(1-\frac{\phi}{\phi_1}) + L_i\frac{\phi}{\phi_1} & 0 \le \phi < \phi_1 \\ L_i & \phi_1 \le \phi < \phi_2 \\ C_{i+1}(1-\frac{1-\phi}{1-\phi_2}) + L_i\frac{1-\phi}{1-\phi_2} & \phi_2 \le \phi < 1 \end{cases}, \tag{2}$$

for the left support foot case and

$$p_i(\phi) = \begin{cases} C_i(1-\frac{\phi}{\phi_1}) + R_i\frac{\phi}{\phi_1} & 0 \le \phi < \phi_1 \\ R_i & \phi_1 \le \phi < \phi_2 \\ C_{i+1}(1-\frac{1-\phi}{1-\phi_2}) + R_i\frac{1-\phi}{1-\phi_2} & \phi_2 \le \phi < 1 \end{cases}, \tag{3}$$

for the right support foot case, where the ϕ is the walk phase t/t_{STEP}, t_{STEP} the duration of the step, C_i the center pose of two feet, and ϕ_1, ϕ_2 the timing parameters determining the transition between single support and double support phase. Also we constrain the boundary condition of the COM position $x_i(\phi)$ as

$$x_i(0) = C_i, \qquad x_i(1) = C_{i+1}. \tag{4}$$

B. Preview Step Generation

For the preview controller, the step controller can use arbitrary sequence of foothold positions. The i_{th} preview step is similarly defined as a set of parameters

$$STEP_i^P = \left\{SF_i, L_i, R_i, L_{1+1}, R_{i+1}, t_{STEP}^i\right\}, \tag{5}$$

where $SF_i \in \{LEFT, RIGHT, DOUBLE\}$ denotes the support foot, L_i, R_i, and L_{i+1}, R_{i+1} are the initial and final 6D poses of left and right foot, and t_{STEP}^i is the duration of the step. Unlike the reactive step controller, arbitrary ZMP trajectory can be used for the preview controller as long as it lies inside the support polygon. For simplicity, we use following piecewise constant ZMP trajectory p_i for this work

$$p_i = \begin{cases} L_i & SF_i = LEFT \\ C_i & SF_i = DOUBLE \\ R_i & SF_i = RIGHT \end{cases}, \tag{6}$$

and these step information is used to update the future ZMP queue $p^{ref}(k)...p^{ref}(k+N-1)$, where k is the current discrete time step and N is the number of preview steps.

IV. Trajectory Controller

A. Analytical COM Trajectory Generation

The piecewise linear ZMP trajectory we use in (2), (3) and boundary conditions (4) yields the following closed-form solution of (7) with zero ZMP error during the step period $0 \leq \phi < 1$

$$x_i(\phi) = \begin{cases} p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} & \\ \quad + m_i t_{ZMP}(\frac{\phi - \phi_1}{\phi_{ZMP}} - \sinh\frac{\phi - \phi_1}{\phi_{ZMP}}) & 0 \leq \phi < \phi_1 \\ p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} & \phi_1 \leq \phi < \phi_2 \\ p_i(\phi) + a_i^p e^{\phi/\phi_{ZMP}} + a_i^n e^{-\phi/\phi_{ZMP}} & \\ \quad + n_i t_{ZMP}(\frac{\phi - \phi_2}{\phi_{ZMP}} - \sinh\frac{\phi - \phi_2}{\phi_{ZMP}}) & \phi_2 \leq \phi < 1 \end{cases} \tag{7}$$

where $\phi_{ZMP} = t_{ZMP}/t_{STEP}$ and m_i, n_i are ZMP slopes which are defined as follows for the left support case

$$m_i = (L_i - C_i)/\phi_1 \tag{8}$$

$$n_i = -(L_i - C_{i+1})/(1 - \phi_2), \tag{9}$$

and for the right support case

$$m_i = (R_i - C_i)/\phi_1 \tag{10}$$

$$n_i = -(R_i - C_{i+1})/(1 - \phi_2). \tag{11}$$

whose parameters a_i^p and a_i^n can then be uniquely determined from the boundary conditions (4). This analytical COM trajectory is continuous and has zero ZMP error during each step period, but can have velocity discontinuity at the transition when commanded velocity is changing.

B. Preview based COM Trajectory Generation

Kajita et. al. proposed a general approximation method to [2] compute the COM trajectory given reference ZMP trajectory based on the following LIPM equation.

$$\ddot{x} = \frac{1}{t_{ZMP}{}^2}(x - p), \tag{12}$$

where $t_{ZMP} = \sqrt{z_0/g}$. If we define a new control variable u_x as the time derivative of the acceleration of COM

$$\frac{\mathrm{d}}{\mathrm{d}t}\ddot{x} = u_x \tag{13}$$

Then we can translate (12) into a strictly proper dynamical system as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x$$

$$p_x = \begin{bmatrix} 1 & 0 & -t_{ZMP}^2 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{14}$$

If we discretize the system of (14) with sampling time of T then

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ p(k) &= Cx(k), \end{aligned} \tag{15}$$

where

$$\begin{aligned} x(k) &\equiv \begin{bmatrix} x(kT) & \dot{x}(kT) & \ddot{x}(kT) \end{bmatrix}^T, \\ u(k) &\equiv u_x(kT), \\ p(k) &\equiv p_x(kT), \\ A &\equiv \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \\ B &\equiv \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}^T, \\ C &\equiv \begin{bmatrix} 1 & 0 & -t_{ZMP}^2 \end{bmatrix}. \end{aligned}$$

Then given the reference ZMP $p^{ref}(k)$, the performance index can be specified as

$$J = \sum_{i=k}^{\infty}\left\{Q_e e(i)^2 + R\Delta u^2(i) + \Delta x^T(i) Q_x \Delta x(i)\right\} \tag{16}$$

where $e(i) \equiv p(i) - p^{ref}(i)$ is ZMP error, $\Delta x(i)$ and $\Delta u(i)$ are the incremental state vector and control input $x(k) - x(k-1)$, $u(k) - u(k-1)$, and Q_e, Q_x and R are weights. If we assume that the ZMP reference p^{ref} can be previewed for N future steps at every sampling time, the optimal controller that minimizes the performance index (16) is given as

$$u(k) = -G_i \sum_{i=0}^{k} e(k) - G_x x(k) - \sum_{j=1}^{N} G_p(j) p^{ref}(k+j), \tag{17}$$

where G_i, G_x and $G_p(j)$ are gains that can be calculated in advance from weights and system parameter of (15).

C. Foot Trajectory Generation

To generate the foot trajectory, we first define the single support walk phase ϕ_{single} as

$$\phi_{single} = \begin{cases} 0 & 0 \leq \phi < \phi_1 \\ \frac{\phi - \phi_1}{\phi_2 - \phi_1} & \phi_1 \leq \phi < \phi_2 \\ 1 & \phi_2 \leq \phi < 1 \end{cases}, \tag{18}$$

then we use following parameterized trajectory function with two heuristic parameters α, β

$$f_T(\phi) = \phi^{\alpha} + \beta\phi(1 - \phi), \tag{19}$$

to generate the foot trajectories for both feet $l_i(\phi)$, $r_i(\phi)$

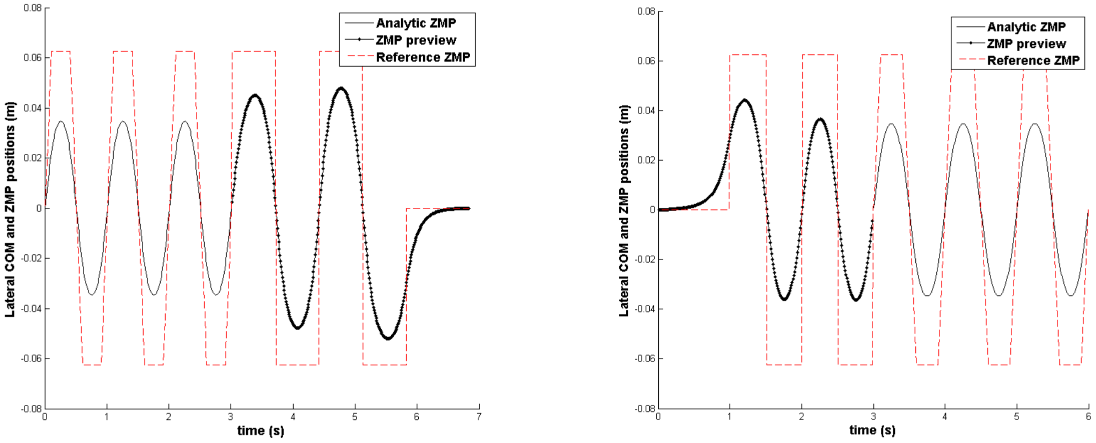
$$l_i(\phi) = L_i(1 - f_T(\phi_{single})) + L_{i+1} f_T(\phi_{single}) \tag{20}$$

$$r_i(\phi) = R_i(1 - f_T(\phi_{single})) + R_{i+1} f_T(\phi_{single}). \tag{21}$$

V. Transition between subcontrollers

A. Transition to Preview Subcontroller

Transition from reactive to preview subcontroller is needed to handle a difficult terrain, which can be straightforwardly



(a) (b)

Fig. 3. The transitions between two trajectory generation subcontrollers. The black line denotes the COM trajectory and the red dashed line denotes the ZMP trajectory. (a) From analytic ZMP subcontroller to ZMP preview subcontroller. (b) From ZMP preview subcontroller to analytic ZMP subcontroller.

done by using the boundary condition of the reactive controller as the initial state of the preview controller. Differentiating (7), we get following boundary values for $x_i(t)$:

$$\begin{aligned}
x_i(t_{STEP}) =& C_{i+1}\\
\dot{x}_i(t_{STEP}) =& \dot{p}_i(t_{STEP}) + \frac{a_i^p e^{1/\phi_{ZMP}} - a_i^n e^{-1/\phi_{ZMP}}}{t_{STEP}\phi_{ZMP}}\\
&+ \frac{n_i t_{ZMP}}{t_{STEP}\phi_{ZMP}}\left(1-\cosh\frac{1-\phi_2}{\phi_{ZMP}}\right)\\
\ddot{x}_i(t_{STEP}) =& \ddot{p}_i(t_{STEP}) + \frac{a_i^p e^{1/\phi_{ZMP}} + a_i^n e^{-1/\phi_{ZMP}}}{t_{STEP}^2\phi_{ZMP}^2}\\
&- \frac{n_i t_{ZMP}}{t_{STEP}^2\phi_{ZMP}^2}\sinh\frac{1-\phi_2}{\phi_{ZMP}} \qquad (22)
\end{aligned}$$

We can further simplify this using (12)

$$\ddot{x}_i(t_{STEP}) = \frac{1}{t_{ZMP}^2}\left(x_i(t_{STEP}) - C_{i+1}\right) = 0 \qquad (23)$$

If we use those values as the initial value of $x(k)$, we can get a continuous COM trajectory up to the second derivative. Figure 3 (a) shows the COM and ZMP trajectories when we switch from reactive subcontroller to preview one. We can see that a longer step duration for preview controller results in a larger amplitude of COM trajectory.

B. Transition to Reactive Subcontroller

Transition from preview subcontroller to reactive subcontroller is not straightforward as the 3D COM trajectory from preview controller may not satisfy the boundary conditions of reactive subcontrollers, which can be derived as

$$\begin{aligned}
x_i(0) =& C_i\\
\dot{x}_i(0) =& \dot{p}_i(0) + \frac{a_i^p - a_i^n + n_i t_{ZMP}\left(1-\cosh\frac{-\phi_2}{\phi_{ZMP}}\right)}{t_{STEP}\phi_{ZMP}}\\
\ddot{x}_i(0) =& \ddot{p}_i(0) + \frac{a_i^p + a_i^n - n_i t_{ZMP}\sinh\frac{-\phi_2}{\phi_{ZMP}}}{t_{STEP}^2\phi_{ZMP}^2} = 0 \qquad (24)
\end{aligned}$$

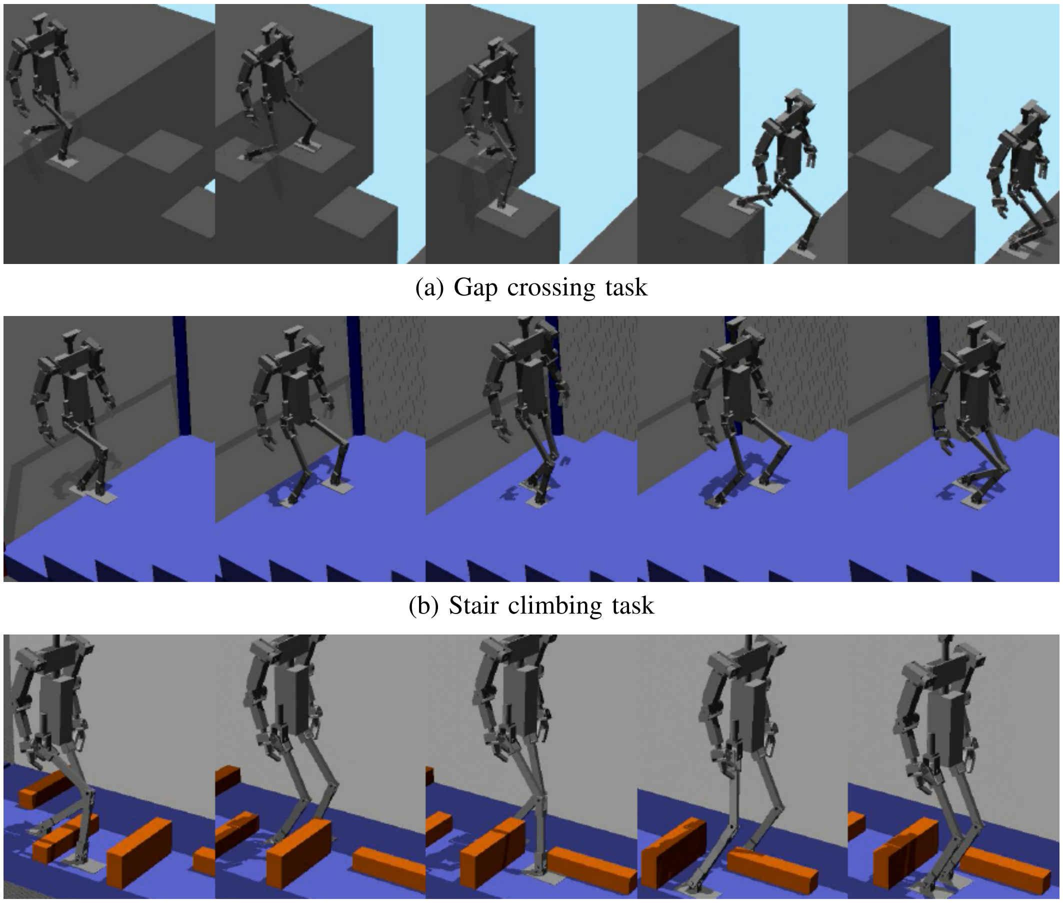
To ensure the preview controller to satisfy the boundary conditions, we add error terms to (16)

$$\begin{aligned}
J =& \sum_{i=k}^{\infty}\left\{Q_e e(i)^2 + R\Delta u^2(i) + \Delta x^T(i) Q_x \Delta x(i)\right\}\\
&+ Q_{t_0}(x(T_{tr}) - x_i(0))^2 + Q_{t_1}(\dot{x}(T_{tr}) - \dot{x}_i(0))^2\\
&+ Q_{t_2}(\ddot{x}(T_{tr}) - \ddot{x}_i(0))^2 \qquad (25)
\end{aligned}$$

where $x(k),\dot{x}(k),\ddot{x}(k)$ are the state of the ZMP preview controller at the discrete time k, $x_i(t),x_i(t),x_i(t)$ the state of analytic controller at continuous time t, Q_{t_0}, Q_{t_1}, Q_{t_2} weights and T_{tr} the discrete time of the transition. In addition to extending the performance index of preview controller, we put N copies of reactive steps with the current walk velocity in the ZMP queue so that the COM trajectory from ZMP preview controller is close to the steady state COM trajectory from reactive controller. We have found that this algorithm generates a continuous and smooth COM trajectories over transitions, as shown in Figure 3 (b).

VI. Simulation Results

The suggested motion controller is implemented on our modular open source humanoid framework [12] which provide a easy adoption to various simulated and physical humanoid platforms. We use the commercial Webots robot simulator [13] based on the Open Dynamics Engine physics library with the simulated model of the THOR full sized humanoid robot we are currently developing for the DARPA Robotics Challenge. Figure 4 shows the simulated THOR robot handling the DARPA Virtual Robotics Challenge (VRC) mobility qualification tasks. The robot is controlled by a low-bandwidth teleoperation at first, and then it initiates the multi-step locomotion to handle the harder terrain utilizing the 3D surface model. The robot is able to walk omnidirectionally with speed up to 1.5 km/h using the reactive walk controller, and it succeeded to cross 80 cm of gap, climb stair with 20 cm step height and step over 20 cm high obstacle using the preview walk controller.



(a) Gap crossing task

(b) Stair climbing task

(c) Obstacle stepping over task

Fig. 4. Simulated THOR robot handling the challenging terrains.

VII. Experimental Results

To demonstrate the advantage of our hybrid walk control approach, we consider a robot soccer scenario where the robot should approach to the ball and kick it. As the ball location may change over time and strong kick induces large momentum, a typical approach is to use a reactive CPG based walk controller to approach the ball, make a full stop and then start a stationary stable kick motion.

Instead of this segmented and stationary stable kicking approach, we use our hybrid walk controller to generate a continuous and dynamically stable kicking motion. When the kick signal is triggered, the walk controller uses the preview subcontroller that uses pre-designed kick step sequence and custom foot trajectory to initiate dynamic kick. After the kick sequence, the walk controller is switched back to reactive subcontroller to keep chasing the ball.

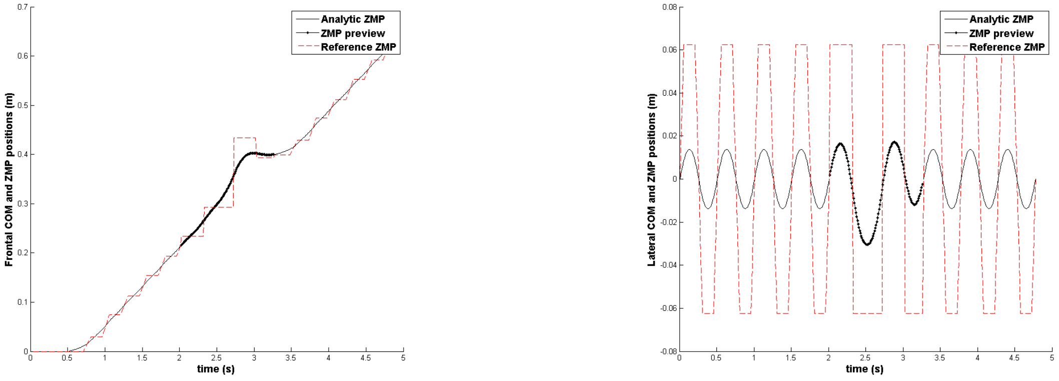
We use a physical DARwIn-OP robot to validate our approach experimentally. The DARwIn-OP robot is 45cm tall, weighs 2.8kg and has position-controlled Dynamixel servos for actuators, which are controlled by a custom microcontroller connected to an Intel Atom-based embedded PC with a control frequency of 100 Hz. Figure 5 shows the generated COM and ZMP trajectories for the robot. We can see the suggested controller smoothly switches between controllers and generate smooth COM trajectory, which is also shown in actual kicking sequences in Figure 6.

VIII. Conclusions

In this work, we describe a hybrid walk controller for a bipedal humanoid robot that can satisfy two conflicting requirements: the ability of reactive stepping and the ability to handle a hard terrain that requires multi-step planning utilizing future stepping positions. Instead of making one walk controller that can satisfy both requirements, we adopt a simpler approach to use two different trajectory generation method and switch them on the fly when needed. Our approach is implemented and demonstrated in physically realistic simulations and experimentally on a DARwIn-OP small humanoid robot. The experimental results show that our method can effectively switch between two trajectory generation methods, which provides the ability to handle hard terrain without sacrificing reactivity. Imminent future work will be implementing current approach to the full sized humanoid robot THOR we are currently building for the DARPA robotics challenge, and the full integration of push recovery algorithms that can benefit from reactive stepping.

Acknowledgments

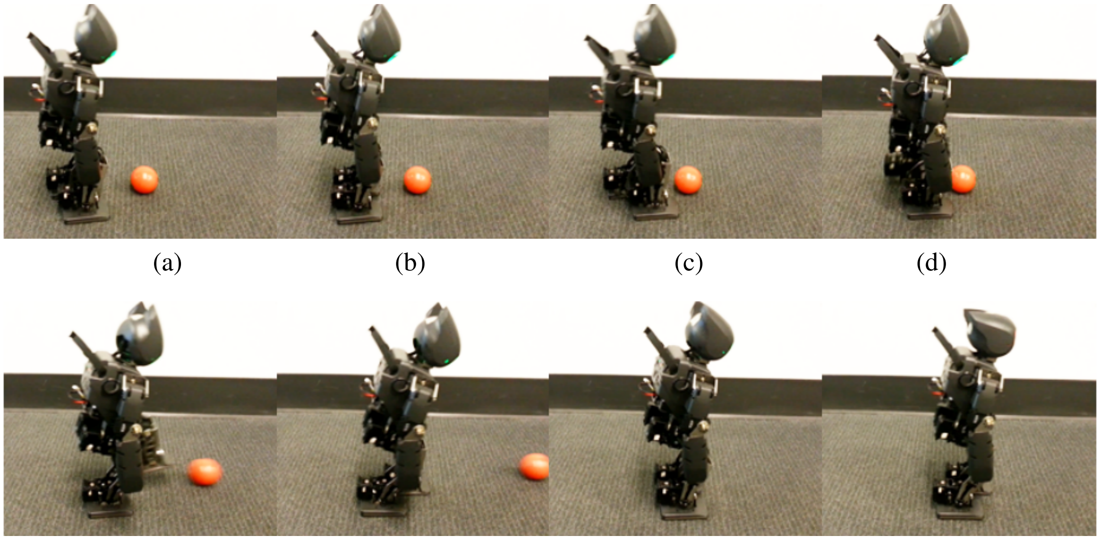
We acknowledge the support of the NSF PIRE program under contract OISE-0730206 and ONR SAFFIR program



(a) Frontal

(b) Lateral

Fig. 5. COM trajectory (black line) and ZMP trajectory (red line) for dynamic kick behavior.



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

(e) (f) (g) (h)

Fig. 6. DARwIn-OP robot performing a dynamic kick in middle of reactive walking.

under contract N00014-11-1-0074. This work was also partially supported by the NRF grant of MEST (0421-20110032), the IT R&D program of MKE/KEIT (KI002138, MARS), and the ISTD program of MKE (10035348).

References

[1] B. Verrelst, O. Stasse, K. Yokoi, and B. Vanderborght, "Dynamically stepping over obstacles by the humanoid robot hrp-2," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, 2006, pp. 117–123.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, and K. H. K. Yokoi, "Biped walking pattern generation by using preview control of zero-moment point," in *in Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 1620–1626.

[3] I. Ha, Y. Tamura, and H. Asama, "Gait pattern generation and stabilization for humanoid robot based on coupled oscillators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3207–3212.

[4] M. Missura and S. Behnke, "Lateral capture steps for bipedal walking," in *IEEE-RAS International Conference on Humanoid Robots*, oct. 2011, pp. 401 –408.

[5] B.-K. Cho, S.-S. Park, and J.-H. Oh, "Stabilization of a hopping humanoid robot for a push," in *IEEE-RAS International Conference on Humanoid Robots*, dec. 2010, pp. 60 –65.

[6] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, "Online learning of a full body push recovery controller for omnidirectional walking," in *IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 1–6.

[7] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method on real-time gait planning for a humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots*, vol. 2, nov. 2004, pp. 640 – 655 Vol. 2.

[8] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, S. Nakaoka, and H. Hirukawa, "A biped pattern generation allowing immediate modification of foot placement in real-time," in *IEEE-RAS International Conference on Humanoid Robots*, dec. 2006, pp. 581 –586.

[9] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *Int. J. Rob. Res.*, vol. 31, no. 11, pp. 1251–1262, Sept. 2012. [Online]. Available: http://dx.doi.org/10.1177/0278364912455720

[10] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 1084–1091.

[11] S.-J. Yi, B.-T. Zhang, D. Hong, and D. D. Lee, "Practical bipedal walking control on uneven terrain using surface learning and push recovery," in *Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 3963–3968.

[12] S. G. McGill, J. Brindza, S.-J. Yi, and D. D. Lee, "Unified humanoid robotics software platform," in *The 5th Workshop on Humanoid Soccer Robots*, 2010.

[13] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004.