2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Ball Walker: A Case Study of Humanoid Robot Locomotion in Non-stationary Environments

Yu Zheng, Katsu Yamane

Abstract— **This paper presents a control framework for a biped robot to maintain balance and walk on a rolling ball. The control framework consists of two primary components: a balance controller and a footstep planner. The balance controller is responsible for the balance of the whole system and combines a state-feedback controller designed by pole assignment with an observer to estimate the system's current state. A wheeled linear inverted pendulum is used as a simplified model of the robot in the controller design. Taking the output of the balance controller, namely the ideal center of pressure of the biped robot on the ball, as the input, the footstep planner computes the foot placements for the robot to track the ideal center of pressure and avoid a fall from the rolling ball. Simulation results show that the proposed controller can enable a biped robot to stably walk on balls of different sizes and rotate a ball to desired positions at desired speeds.**

I. INTRODUCTION

Nowadays, we can see many humanoid robots that successfully walk on various terrains, such as slopes, stairs, and even rough terrains. Most often, however, these environments are stationary and unchanged during the motion of a robot. Up to now, the study of humanoid locomotion in dynamic environments is still limited. Kuroki et al. [1], [2], Hyon [3], and Anderson and Hodgins [4] developed motion or force control systems such that a biped robot can maintain balance on a moving platform or under external forces. In this paper, we focus on a more challenging case of biped locomotion in non-stationary environments, i.e., biped walking on a rolling ball. Differently from the previous work [1]–[4], not only the balance controller but also the stepping strategy on the ball need to be considered in our case. We propose a control framework for this motion (Fig. 1), which consists of a balance controller and a footstep planner.

The balance controller is responsible for maintaining balance of the whole system and produces the ideal center of pressure (CoP) for the robot on the ball. It combines a state-feedback controller designed by pole assignment with an observer to estimate the current state of the robot. A wheeled linear inverted pendulum is adopted as a simplified dynamics model of the robot, for which a stable balance controller can be easily designed.

Taking the ideal CoP generated by the balance controller as an input, the footstep planner computes the foot placement of the biped robot on the ball. The actual CoP may be different from the ideal CoP because the CoP must be in the contact region. Therefore, the time to lift up or touch down a foot and the velocity of a swing foot are elaborately chosen such that the actual CoP can track the ideal CoP as much as possible.

Yu Zheng is with the Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599, USA. This work was done when he was with Disney Research Pittsburgh, PA 15213, USA. yuzheng001@gmail.com

Katsu Yamane is with Disney Research Pittsburgh, PA 15213, USA. kyamane@disneyresearch.com

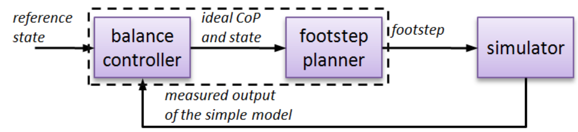


Fig. 1. Overview of the controller (dashed line box).

We demonstrate our controller with many illustrative simulations. Various reference state inputs are applied to the balance controller and the controller is able to make a rolling ball reach different desired positions or speeds. All simulations are carried out on balls of various sizes to verify the applicability of our controller to different cases.

This paper is organized as follows. Section II summarizes related work. Sections III and IV introduce the balance controller and its application to the velocity control of the rolling ball, respectively. Section V explains our footstep planner. Simulation results and discussions are given in Section VI. Section VII concludes with future work.

II. RELATED WORK

A. Biped Locomotion Generation and Footstep Planning

In generating the walking pattern for a biped robot, it is often assumed that the footsteps are given, and the problem becomes how to compute the joint angles such that the resulting motion satisfies the balance condition and the given footsteps. Based on the given footsteps, one may first determine a CoP trajectory and then a physically consistent trajectory of the center of mass (CoM) of the robot using a simplified model, such as an inverted pendulum [5], [6]. Thus the joint angles of the robot can be computed using inverse kinematics according to the CoM trajectory and the footsteps. This technique has been successfully used to generate biped walking pattern in a stationary environment [7], [8]. In an environment with known obstacles, several algorithms have been proposed to compute footsteps and a collision-free path for a biped robot [9]–[11], such that the walking pattern can be generated later in this way. The stepping motion can also be predefined in motion capture data when a robot is required to track a captured motion [12]. However, the method based on predefined footsteps may not be suitable for generating motions in a non-stationary environment, such as on a rolling ball, because footsteps usually cannot be predetermined in that situation.

Humanoid locomotion can also be generated in a reverse way. One may first have a reference or desired body motion for a biped robot, such as captured motions from humans or other characters, and command the robot to realize that motion [13]. In this case, the required CoP trajectory can be computed from the reference body motion based on a simplified robot model. Then, appropriate footsteps that satisfy the computed CoP trajectory need to be determined such that the joint angles can be calculated likewise using inverse kinematics. Unfortunately, there is not much work on this kind of approaches [14].

Footstep planning is also considered in humanoid push recovery [15]–[18]. Undergoing a high magnitude push, a robot might lose balance and need to take several steps to recover. Those footsteps must be carefully determined in order for the robot to avoid a fall. Nevertheless, a push applies only an instantaneous disturbance to the robot. Differently from this situation, a robot walking on a rolling ball has to sustain continuous dynamic interaction with the ball.

B. Robot Control in a Non-stationary Environment

So far, the study of robot motion control in non-stationary environments is still limited. Kuroki et al. [1], [2] proposed a motion control system to maintain balance of a small biped robot on a moving platform or under external forces. For the same purpose, Hyon [3] presented a contact force control framework for the balance control of a human-size robot on rough terrain under external forces, while Anderson and Hodgins [4] developed methods for adapting models of humanoid robots performing dynamic tasks. In those cases, however, the robot's feet keep stationary contact with the platform or ground and no stepping motion is involved.

Besides humanoid robots, some other robots may work in a non-stationary environment, such as multi-wheeled robots balancing on and driving a ball [19]–[21]. In that case, the wheels always make three or four symmetric contacts with the ball, which greatly benefits the balance control of the robot. However, the feet of a biped robot can only make one or two contacts with the ball, which are usually asymmetrical about the top of the ball. Furthermore, because of the limited foot size and support region, the ideal CoP, which is continuously changing on the rolling ball, may go beyond the support region. Hence, we have to not only design controllers to maintain system's balance but also combine them with footstep planning to provide the robot with timely support on a rolling ball.

III. Balance Controller

Referring to the balance controller presented in [13], we design the balance controller for biped walking on a rolling ball (see Fig. 2). Similarly to [13], our balance controller comprises a state-feedback controller and an observer. The state-feedback controller is designed by pole assignment for a simplified model of the robot and determines the input to the model to maintain its balance. The observer estimates the current state based on the estimated and measured output of the model. In this section, we start with the introduction of the simplified dynamics model, followed by the details of the balance controller.

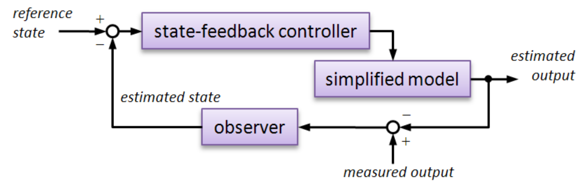


Fig. 2. Overview of the balance controller.

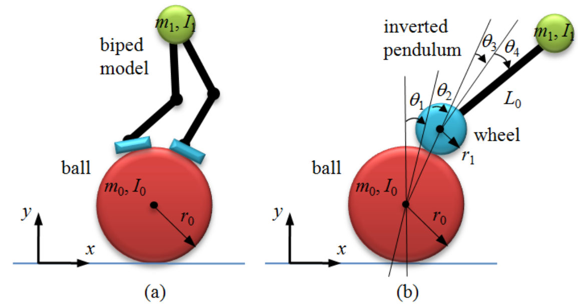


Fig. 3. A biped robot simplified as a wheeled inverted pendulum.

A. Simplified Dynamics Model

We use a linear inverted pendulum with a massless wheel as the simplified dynamics model of a biped robot on a ball, as illustrated in Fig. 3. Let r_0, m_0, and I_0 denote the radius, the mass, and the inertia of the ball, respectively, r_1 the radius of the wheel, m_1 and I_1 the mass and the inertia of the inverted pendulum, respectively, and L_0 the distance from the wheel center to the mass center of the inverted pendulum. The angle θ_1 indicated in Fig. 3 is the roll angle of the ball, θ_2 and θ_3 represent the relative position and rotation of the wheel on the ball and satisfy the condition $\theta_3 = r_0\theta_2/r_1$ to avoid relative slippage at the contact, and θ_4 indicates the swing of the inverted pendulum relative to the wheel. Thus, the model has three free variables, namely θ_1, θ_2, and θ_4. The positive direction of angles is taken to be clockwise.

The motion equation of the model is written as

$$\boldsymbol{M}\ddot{\boldsymbol{\theta}} + \boldsymbol{G}\boldsymbol{\theta} = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \theta_4]^T$, $\boldsymbol{\tau} = [0 \quad \tau_2 \quad 0]^T$, τ_2 is the joint torque corresponding to θ_2, and

$$\boldsymbol{M} = \begin{bmatrix} m_0 r_0^2 + m_1 L_1^2 + I & m_1 L_1 L_2 + k I_1 & m_1 L_0 L_1 + I_1 \\ m_1 L_1 L_2 + k I_1 & m_1 L_2^2 + k^2 I_1 & m_1 L_0 L_2 + k I_1 \\ m_1 L_0 L_1 + I_1 & m_1 L_0 L_2 + k I_1 & m_1 L_0^2 + I_1 \end{bmatrix}$$

$$\boldsymbol{G} = -\begin{bmatrix} m_1 g (L_0 + L_3) & m_1 g L_2 & m_1 g L_0 \\ m_1 g L_2 & m_1 g (k^2 L_0 + L_3) & m_1 g k L_0 \\ m_1 g L_0 & m_1 g k L_0 & m_1 g L_0 \end{bmatrix}$$

$$L_1 = 2r_0 + r_1 + L_0, \quad L_2 = r_0 + r_1 + k L_0$$

$$L_3 = r_0 + r_1, \quad I = I_0 + I_1, \quad k = 1 + r_0/r_1.$$

B. Details of the Controller

The motion equation (1) can be rewritten as the following state-space differential equation:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u \tag{2}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \tag{3}$$

where $\boldsymbol{x} = [\boldsymbol{\theta}^T \quad \dot{\boldsymbol{\theta}}^T]^T$ is the state, $u = \tau_2$ is the input, $\boldsymbol{y} = \boldsymbol{\theta}$ is the output of the simplified model, and the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are given by

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \\ -\boldsymbol{M}^{-1}\boldsymbol{G} & \boldsymbol{0}_{3\times3} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_{3\times1} \\ \boldsymbol{M}^{-1}\boldsymbol{b} \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T.$$

We also design a state-feedback controller as

$$u = \boldsymbol{K}(\boldsymbol{x}^* - \boldsymbol{x}) \tag{4}$$

where $\boldsymbol{K} \in \mathbb{R}^{1\times6}$ is the feedback gain and $\boldsymbol{x}^*$ is the target state. In this paper, $\boldsymbol{x}^*$ is computed from a target roll angle θ_1^* of the ball by $\boldsymbol{x}^* = \boldsymbol{T}\theta_1^*$, where $\boldsymbol{T} = [1 \ \ -1 \ \ r_0/r_1 \ \ 0 \ \ 0 \ \ 0]^T$ maps the target roll angle θ_1^* to the corresponding equilibrium state where $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{0}$. Also, the feedback gain $\boldsymbol{K}$ is chosen so that $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}$ is stable; i.e., all of its eigenvalues have negative real parts. By doing this, the ball rolls and stops at any given desired roll angle θ_1^*.

Because we do not have access to the real state, we replace the state $\boldsymbol{x}$ in (2) and (4) with its estimate $\hat{\boldsymbol{x}}$ and design an observer that compares the estimated and actual outputs to update the estimated state $\hat{\boldsymbol{x}}$ as

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}u + \boldsymbol{F}(\boldsymbol{y} - \hat{\boldsymbol{y}}) \tag{5}$$

where $\boldsymbol{F} \in \mathbb{R}^{6\times3}$ is the observer gain, $\hat{\boldsymbol{y}} = \boldsymbol{C}\hat{\boldsymbol{x}}$ is the estimated output, and $\boldsymbol{y}$ is the measured output.

Combining (2)–(5), we obtain the following system of the estimated state and new input $\boldsymbol{u}_c = [\boldsymbol{y}^T \quad \theta_1^*]^T$:

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}_c\hat{\boldsymbol{x}} + \boldsymbol{B}_c\boldsymbol{u}_c \tag{6}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{C}_c\hat{\boldsymbol{x}} \tag{7}$$

where

$$\boldsymbol{A}_c = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K} - \boldsymbol{F}\boldsymbol{C}, \quad \boldsymbol{B}_c = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{B}\boldsymbol{K}\boldsymbol{T} \end{bmatrix}, \quad \boldsymbol{C}_c = \boldsymbol{C}.$$

The sum of the angles $\hat{\theta}_1$ and $\hat{\theta}_2$ generated by the balance controller indicates the contact location between the wheel and the ball, which will be used as the ideal CoP of the robot on the ball and an input to the footstep planner.

IV. Constant Velocity Control

In this section, we address a specific formulation of the target roll angle θ_1^*, which enables the controller to generate the motion of a ball with a desired constant (average) velocity. To simplify the notations, herein we omit the observer in the controller and assume that the actual state $\boldsymbol{x}$ and output $\boldsymbol{y}$ in (6) are equal to their estimates $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$, respectively.

We begin with the definition of notations. Let $\boldsymbol{K}$ and $\boldsymbol{T}$ be partitioned respectively into two subvectors with three components as

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_1^T & \boldsymbol{K}_2^T \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_1^T & \boldsymbol{0}_{3\times1}^T \end{bmatrix}^T.$$

Note that the matrices $\boldsymbol{A}$, $\boldsymbol{K}$, and $\boldsymbol{T}$ have the properties $\boldsymbol{A}\boldsymbol{T} = \boldsymbol{0}_{6\times1}$ and $\boldsymbol{K}\boldsymbol{T} = \boldsymbol{K}_1^T\boldsymbol{T}_1$. Also we define

$$\boldsymbol{T}' = \begin{bmatrix} \boldsymbol{0}_{3\times1}^T & \boldsymbol{T}_1^T \end{bmatrix}^T.$$

Note that $\boldsymbol{A}\boldsymbol{T}' = \boldsymbol{T}$ and $\boldsymbol{K}\boldsymbol{T}' = \boldsymbol{K}_2^T\boldsymbol{T}_1$.

Now we consider a reference trajectory for θ_1 with a constant velocity $\dot{\theta}_1^r$. Without loss of generality, we can write the reference θ_1 as $\theta_1^r = \dot{\theta}_1^r t$. We expect $\boldsymbol{\theta} = \boldsymbol{T}_1\theta_1^r$ so that $\dot{\boldsymbol{\theta}} = \boldsymbol{T}_1\dot{\theta}_1^r$ and $\ddot{\boldsymbol{\theta}} = \boldsymbol{0}$, and particularly $\dot{\theta}_1 = \dot{\theta}_1^r$ and $\ddot{\theta}_1 = 0$. From (6) without the observer, we obtain

$$\boldsymbol{M}\ddot{\boldsymbol{\theta}} = \boldsymbol{b}\boldsymbol{K}_1^T\boldsymbol{T}_1\theta_1^* - \boldsymbol{G}\boldsymbol{\theta} - \boldsymbol{b}(\boldsymbol{K}_1^T\boldsymbol{\theta} + \boldsymbol{K}_2^T\dot{\boldsymbol{\theta}}). \tag{8}$$

If $\boldsymbol{\theta} = \boldsymbol{T}_1\theta_1^r$, then $\ddot{\boldsymbol{\theta}} = \boldsymbol{0}$ and $\boldsymbol{G}\boldsymbol{\theta} = \boldsymbol{0}$. Substituting them into (8) yields

$$\theta_1^* = \theta_1^r + \frac{\boldsymbol{K}_2^T\boldsymbol{T}_1}{\boldsymbol{K}_1^T\boldsymbol{T}_1}\dot{\theta}_1^r = \theta_1^r + \frac{\boldsymbol{K}\boldsymbol{T}'}{\boldsymbol{K}\boldsymbol{T}}\dot{\theta}_1^r. \tag{9}$$

We shall prove that by using the input given by (9), $\boldsymbol{\theta}$ will converge to $\boldsymbol{T}_1\theta_1^r$ and $\dot{\boldsymbol{\theta}}$ to $\boldsymbol{T}_1\dot{\theta}_1^r$. To do this, let us consider the error between $\boldsymbol{T}\theta_1^r + \boldsymbol{T}'\dot{\theta}_1^r$ and $\boldsymbol{x}$; i.e.,

$$\boldsymbol{e} = \boldsymbol{T}\theta_1^r + \boldsymbol{T}'\dot{\theta}_1^r - \boldsymbol{x}. \tag{10}$$

From (6), (9), and (10) we have

$$\begin{aligned}\dot{\boldsymbol{e}} &= \boldsymbol{T}\dot{\theta}_1^r - \dot{\boldsymbol{x}} \\ &= \boldsymbol{T}\dot{\theta}_1^r - \boldsymbol{B}\boldsymbol{K}\boldsymbol{T}\theta_1^* - (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x} \\ &= -\boldsymbol{B}\boldsymbol{K}\boldsymbol{T}\theta_1^r + (\boldsymbol{T} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{T}')\dot{\theta}_1^r - (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}.\end{aligned}$$

By adding $\boldsymbol{A}\boldsymbol{T}\theta_1^r (= \boldsymbol{0}_{6\times1})$ to the above equation and replacing $\boldsymbol{T}$ in the second term with $\boldsymbol{A}\boldsymbol{T}'$, we can derive

$$\dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{e}. \tag{11}$$

Because $\boldsymbol{K}$ is chosen so that the real parts of all eigenvalues of $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}$ are negative, $\boldsymbol{e}$ converges to $\boldsymbol{0}$ asymptotically. Moreover, from (10) it follows that

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{T}_1\theta_1^r - \boldsymbol{\theta} \\ \boldsymbol{T}_1\dot{\theta}_1^r - \dot{\boldsymbol{\theta}} \end{bmatrix}. \tag{12}$$

Therefore, $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ converge to $\boldsymbol{T}_1\theta_1^r$ and $\boldsymbol{T}_1\dot{\theta}_1^r$, respectively, and particularly $\dot{\theta}_1$ converges to $\dot{\theta}_1^r$. As a consequence, we may take $\dot{\theta}_1^r$ to be a desired rolling velocity of the ball and apply the input θ_1^* given by (9) to the balance controller to achieve that velocity.

V. Footstep Planner

A. Motion of the Supporting Foot

Assume that each robot's foot has a flat sole and can rotate on the ball without slippage. As shown in Fig. 4(a), we use an angle θ_c to express the location of the center of a foot relative to the ball and another angle θ_d to express the rotation of a foot from its center. Due to the limited foot length, the foot

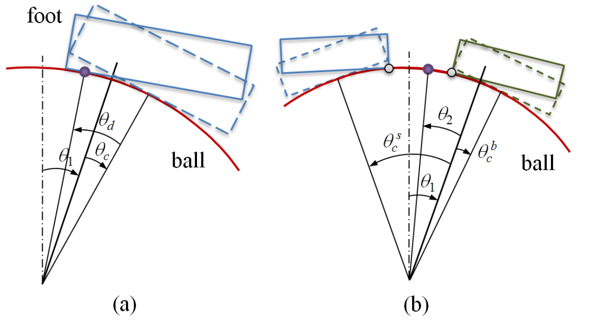


Fig. 4. Side view of feet on a ball. θ_1 is the roll angle of the ball. θ_c indicates the center of the foot relative to the ball, while $\theta_c + \theta_d$ indicates the contact position between the foot and the ball. (a) The contact point is the actual CoP (solid dot) in the single support phase. (b) The CoP (solid dot) can lie between the contacts (white dots) of two feet with the ball during double support.

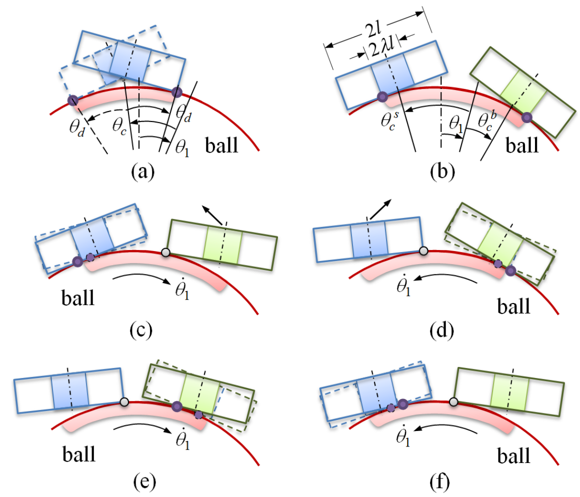


Fig. 5. Foot motion on a rolling ball. The solid and white dots represent the CoP and the contact between a foot and the ball, respectively. (a),(b) Single and double support domains. A reduced contact area is used to determine the double support domain. The allowable domain of the CoP is marked in pink. (c),(d) θ_2 goes beyond the double support domain, so the foot on the right or left side starts to swing. (e),(f) The CoP lies in the allowable domain and there is no need to move a foot.

rotation on the ball is also limited and θ_d must be within the domain $[-l/r_0, l/r_0]$, where l is the half foot length. Each foot is associated with such a couple of angles θ_c and θ_d, by which the position and orientation of the foot on the ball is uniquely determined. After a foot comes into contact with the ball, θ_c for the foot is determined and unchanged until next time the foot shifts to a new location (how to determine the value of θ_c will be addressed in the next subsection), while θ_d for the foot starts to change along with the value $\hat{\theta}_2$ produced by the balance controller as

$$\theta_d = \begin{cases} \hat{\theta}_2 - \theta_c & \text{if } |\hat{\theta}_2 - \theta_c| \le l/r_0 \\ \mathrm{sign}(\hat{\theta}_2 - \theta_c) l/r_0 & \text{otherwise} \end{cases}. \tag{13}$$

The sum $\theta_c + \theta_d$ indicates the contact location between the foot and the ball, which we call the actual CoP. From the controller we obtain the angle $\hat{\theta}_2$, which specifies the ideal CoP. In the single support phase, $\theta_c + \theta_d$ for the supporting foot gives the actual CoP and the actual θ_2. We expect $\theta_c + \theta_d$ to be equal to $\hat{\theta}_2$ with $|\theta_d| \le l/r_0$, or equivalently $\hat{\theta}_2$ to be within the domain $[\theta_c - l/r_0, \theta_c + l/r_0]$, which is called the *single support domain*. This implies that the actual θ_2 can reach its ideal value $\hat{\theta}_2$ and the supporting foot can reach the ideal CoP, as depicted in Fig. 4(a). Fig. 5(a) shows the domain of the CoP that can be achieved by the single supporting foot.

In the double support phase, the actual CoP can lie in the domain between the contacts of two feet with the ball, as shown in Fig. 4(b). Accordingly, the feasible domain of the actual θ_2 can be $[\theta_c^s - l/r_0, \theta_c^b + l/r_0]$ and called the *double support domain*, where θ_c^s and θ_c^b are the smaller and bigger θ_c of the feet, as indicated in Fig. 4(b). In practice, however, we shrink the contact domain on each foot during double support to $[\theta_c - \lambda l/r_0, \theta_c + \lambda l/r_0]$, where $\lambda \in [0, 1)$. Then the allowable double support domain for the actual θ_2 is reduced to $[\theta_c^s - \lambda l/r_0, \theta_c^b + \lambda l/r_0]$, and the allowable domain for the CoP is also reduced, as illustrated in Fig. 5(b). Once the actual θ_2 exceeds this reduced domain, one foot starts to lift up and swing to a new contact position, but the actual θ_2 can still be maintained within the single support domain of the remaining supporting foot and track the ideal value $\hat{\theta}_2$ given by the balance controller. Also, the actual CoP can follow the ideal CoP. Therefore, we have the following condition for determining the time to swing a foot:

Swing condition: If $\theta_d < -\lambda l/r_0$ for both supporting feet, then the foot with bigger θ_c starts to swing, as depicted in Fig. 5(c). If $\theta_d > \lambda l/r_0$ for them, then the foot with smaller θ_c starts to swing, as shown in Fig. 5(d).

If the angles θ_d for both supporting feet decrease to values smaller than $-\lambda l/r_0$ [Fig. 5(c)], this implies that the actual θ_2 is decreasing, and thus the foot with $\theta_c = \theta_c^b$ should move to reduce its θ_c. In this case, the relation $\theta_c^s + \theta_d^s = \theta_2 \le \theta_c^b + \theta_d^b$ holds, since the ideal CoP is still reached by the other supporting foot, for which $\theta_c = \theta_c^s$. Similarly, in the case where both θ_d become bigger than $\lambda l/r_0$ [Fig. 5(d)], the actual θ_2 is increasing, and $\theta_c^s + \theta_d^s \le \theta_2 = \theta_c^b + \theta_d^b$. Hence, the foot with smaller θ_c should move to increase it. When none of the two situations occurs, the actual θ_2 must be within the double support domain and no foot needs to move, as depicted in Fig. 5(e) and (f).

B. Motion of the Swing Foot

We tried various ways to determine the motion of a swing foot and its next location on the ball, such as setting a fixed step length or swing speed or always landing the swing foot on the top of the ball, in order to make the actual CoP track the ideal CoP as much as possible. Herein we introduce a simple and effective approach, by which we achieved very stable walking behaviors of a biped on balls of different sizes.

The general idea is to change the angle θ_c of the swing foot ahead of the variation of $\hat{\theta}_2$. Before $\hat{\theta}_2$ exceeds the single support domain, the swing foot needs to make a new contact with the ball such that $\hat{\theta}_2$ can be kept within the new double support domain. We express the instantaneous velocity of the swing foot relative to the ball by $\dot{\theta}_c$. First, intuitively, $\dot{\theta}_c$ should be proportional to $\dot{\hat{\theta}}_2$, which enables the swing foot to track the variation of $\hat{\theta}_2$. Secondly, from Fig. 3 we see that $\theta_1 + \theta_2$ gives the position of the CoP relative to the top of the ball. When $\theta_2 = -\theta_1$, the CoP is on the top of the ball. Thus $-\dot{\hat{\theta}}_1$ indicates the motion direction for the swing foot to reach the top of the ball. From these arguments, we set the instantaneous swing velocity as

$$\dot{\theta}_c = \begin{cases} -k_1\dot{\hat{\theta}}_1 + k_2\dot{\hat{\theta}}_2 & \text{if } |-k_1\dot{\hat{\theta}}_1 + k_2\dot{\hat{\theta}}_2| \le \dot{\theta}_c^U \\ \operatorname{sign}(-k_1\dot{\hat{\theta}}_1 + k_2\dot{\hat{\theta}}_2)\dot{\theta}_c^U & \text{otherwise} \end{cases} \tag{14}$$

where k_1 and k_2 are nonnegative scalars and $\dot{\theta}_c^U$ is the upper bound of $\dot{\theta}_c$. The coefficients $-k_1$ and k_2 scale the velocities $\dot{\hat{\theta}}_1$ and $\dot{\hat{\theta}}_2$, which enables the swing foot to remain close to the top of the ball and track the ideal CoP. Because of the mechanical limitation of a robot, the swing velocity of a foot cannot be arbitrarily large and must have an upper bound. With the velocity given by (14), the angle θ_c for the swing foot is simply updated by

$$\theta_c = \theta_c + \dot{\theta}_c \Delta t \tag{15}$$

where Δt is the time step.

The time to land a swing foot on the ball depends on several factors. We use a hybrid condition as below:

Landing condition: A swing foot touches down if (a) the swing time exceeds a given time interval t_s or the angle $|\hat{\theta}_1 + \theta_c|$ for either the swing or supporting foot exceeds a given limit α and (b) $(\hat{\theta}_2 - \theta_c^2)(\theta_c^1 - \theta_c^2) > k_3(\hat{\theta}_2 - \theta_c^2)^2$, where θ_c^1 and θ_c^2 are the angles θ_c of the swing and supporting feet, respectively, and k_3 is a nonnegative scalar.

First, we usually do not wish a swing foot to touch down too quickly and thus set a time threshold t_s. Second, for the sake of limited friction and the purpose of safety, we expect both feet not to move far away from the top of the ball. Hence, in condition (a) we also restrict the angle $|\hat{\theta}_1 + \theta_c|$, which indicates the position of a foot relative to the top of the ball. Condition (b) compares the distance between the swing and supporting feet, which is measured by $\theta_c^1 - \theta_c^2$, with the deviation of the ideal CoP from the center of the supporting foot, which is measured by $\hat{\theta}_2 - \theta_c^2$. It helps to ensure that $\hat{\theta}_2$ lies in the new double support domain and the ideal CoP is between two feet after the swing foot touches down. If both conditions (a) and (b) are satisfied, then the swing foot touches down; otherwise it can continue swinging. Once the swing foot touches down, its angle θ_d is computed by (13).

VI. Simulation Results and Discussions

A. Simulator

The footstep planner attempts to compute a footstep sequence so that the contact region includes the ideal CoP as much as possible. However, it may exceed the boundary of the supporting foot before the swing foot touches down, in which case, the actual CoP deviates from the ideal CoP. In order to test the capability of the footstep planner to track the ideal CoP and the influence of the CoP difference on balance control, we design a simulator based on the same simplified model as used in the balance controller. The only difference is that the input here is $\ddot{\theta}_2$ because τ_2 does not exist in the biped model.

We obtain $\ddot{\theta}_2$ by differentiating the actual θ_2 that can be computed from the actual CoP. In single support phase, the actual CoP is the unique contact point between the supporting foot and the ball. In double support phase, if the ideal CoP is out of the contact region, we assume that the actual CoP is the contact point closer to the ideal CoP. If the ideal CoP is in the contact region, on the other hand, we assume that the ideal CoP can be realized by adjusting the joint torques.

By left multiplying both sides of (1) with $\boldsymbol{M}^{-1}$ and some matrix manipulation, we obtain

$$\ddot{\theta}_2 + \boldsymbol{m}_2 \boldsymbol{G} \boldsymbol{\theta} = m_{22}\tau_2 \tag{16}$$

where $\boldsymbol{m}_2$ and m_{22} are the 2-nd row and the (2,2)-th entry of $\boldsymbol{M}^{-1}$, respectively. From (16) it follows that τ_2 for the simplified model in the simulator is

$$\tau_2 = \frac{1}{m_{22}}(\ddot{\theta}_2 + \boldsymbol{m}_2 \boldsymbol{G} \boldsymbol{\theta}). \tag{17}$$

Substituting (17) into (1), we have the following state-space differential equation to describe the motion of the simplified model in the simulator:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_s \boldsymbol{x} + \boldsymbol{B}_s \boldsymbol{u}_s \tag{18}$$

$$\boldsymbol{y} = \boldsymbol{C}_s \boldsymbol{x}. \tag{19}$$

where $\boldsymbol{u}_s = \ddot{\theta}_2$ and

$$\boldsymbol{A}_s = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \\ -\boldsymbol{M}^{-1}\boldsymbol{G} + \frac{1}{m_{22}}\boldsymbol{m}_2^T\boldsymbol{m}_2 & \boldsymbol{0}_{3\times3} \end{bmatrix}, \ \boldsymbol{B}_s = \begin{bmatrix} \boldsymbol{0}_{3\times1} \\ \frac{1}{m_{22}}\boldsymbol{m}_2^T \end{bmatrix}$$

$$\boldsymbol{C}_s = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}.$$

Here, $\boldsymbol{x}$ and $\boldsymbol{y}$ have the same components and initial values as those in the balance controller. The value $\boldsymbol{y}$ from (19) is the measured output, which is a part of the input to the balance controller (see Figs. 1 and 2).

B. Simulation Setup

The simplified models in the balance controller and the simulator have the same parameters: $m_0 = 40$ kg, $I_0 = 0.5\ \mathrm{kg \cdot m^2}$, $m_1 = 65$ kg, $I_1 = 0.5\ \mathrm{kg \cdot m^2}$, $r_1 = 0.1$ m, and $L_0 = 0.8$ m. The ball in the simulation may have four sizes: $r_0 = 0.25$, 0.5, 0.75, and 1.0 m. It has been verified that both the controllability matrix and the observability matrix have full rank, which implies that the system is controllable and observable.

The pole values for determining the state-feedback controller gain $\boldsymbol{K}$ in (4) and the observer gain $\boldsymbol{F}$ in (5) are $[-24.9 \quad -25 \quad -4.55 \quad -4.5 \quad -1.05 \quad -1]$ and $[-49.8 \ -50 \ -4.55 \ -4.5 \ -1.05 \ -1]$, respectively. The pole values for the state-feedback controller are elaborately chosen such that the ball rolls at a reasonable speed and the biped robot on the ball can follow that speed.

The parameters of the robot's feet used in the footstep planner are $l = 0.1$ m and $\lambda = 0.2$. The scales k_1 and k_2

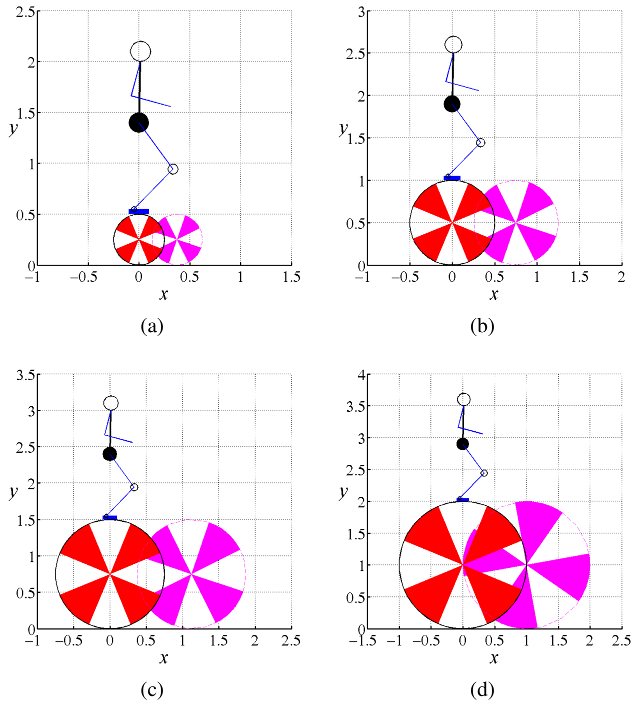


Fig. 6. The initial pose of the robot on the ball. The target position of the ball is marked in magenta.

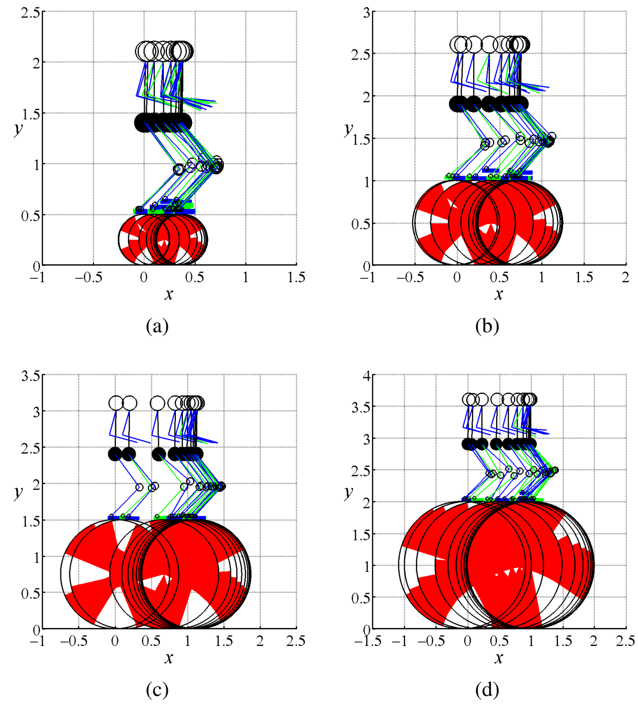


Fig. 7. Snapshots of the resulting motions.

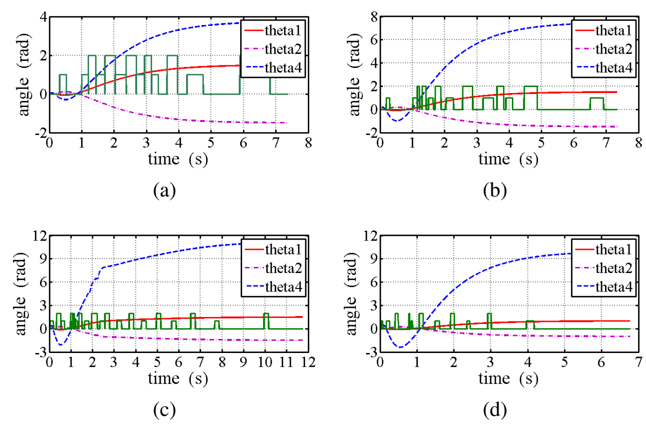


Fig. 8. Measured output of the simplified model in the simulator. The green line shows the foot swing, where the zero y-coordinate means the double support and different nonzero y-coordinates represent different feet.

in (14) to determine the swing velocity are both taken to be 0.2. The minimum swing time t_s is taken to be 0.2 s. Then the upper bound $\dot{\theta}_2^U$ of $\dot{\theta}_2$ in (14) is chosen as $0.5l/r_0t_s$, $2l/r_0t_s$, $4l/r_0t_s$, and $8l/r_0t_s$ rad/s for $r_0 = 0.25$, 0.5, 0.75, and 1.0, respectively. On a bigger ball, we allow the robot to swing feet faster because the instantaneous velocity at the top of a bigger ball is larger than of a smaller ball when the balls roll at the same angular velocity or have the same $\dot{\theta}_1$.

C. Simulation with a Constant Input

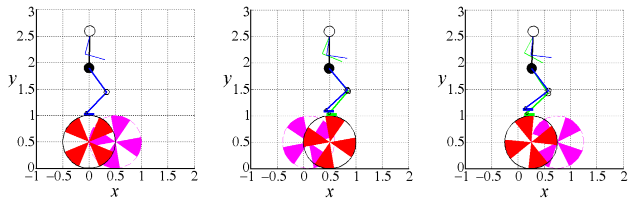
We start the simulation with a simple case in which a single constant input, namely the desired roll angle θ_1^* of the ball, is applied to the controller. Fig. 6 shows the initial and desired positions of different balls. The resulting motion is exhibited with snapshots in Fig. 7 and the accompanying video. It can be seen that the robot can walk stably on the balls and move them to the desired positions. Fig. 8 plots the measured output of the simplified model from the simulator and the planned footsteps.

We also test our controller in the case where the input suddenly changes at certain time frames when the robot is still walking, as depicted in Fig. 9. Likewise, this test is conducted on balls of different sizes. The accompanying video shows that the balance controller well maintains the robot's balance and the footstep planner computes the proper footsteps after the input changes, which together enable the robot to successfully walk on the rolling balls.

D. Simulation with Constant Velocity Control

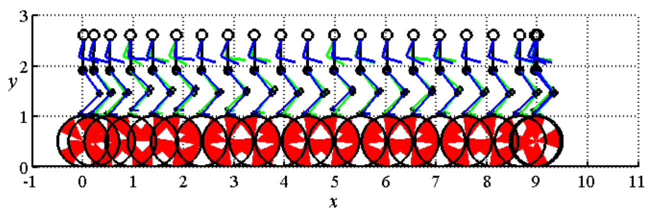
Now we try the specific input given by (9) for the constant velocity control. The desired rolling velocity $\dot{\theta}_1^r = 0.4$ rad/s. It should be noted that the difference between the target roll angle θ_1^* from (9) and the state value θ_1 in the controller is increasing at the beginning of motion, since $\dot{\theta}_1 < \dot{\theta}_1^r$. This could result in a very large acceleration $\ddot{\theta}_1$ during the initial period and cause $\dot{\theta}_1$ to exceed the desired velocity $\dot{\theta}_1^r$. A large overshoot of velocity may incur the failure of the footstep planning because the robot's feet cannot move faster to follow the rolling of the ball. In order to avoid a large overshoot and have a gentle increase in the rolling velocity, we arbitrarily set several intermediate reference velocities $\dot{\theta}_1^r = 0.2, 0.3, 0.35$, and 0.375 rad/s. Once $\dot{\theta}_1$ reaches $\dot{\theta}_1^r - \epsilon$, $\dot{\theta}_1^r$ changes to the next intermediate value, where ϵ is the tolerance and taken to be 0.01 here. We conduct the test on balls of radius 0.5 m and 1.0 m, respectively.

Fig. 10 displays the snapshots of the resulting motions.

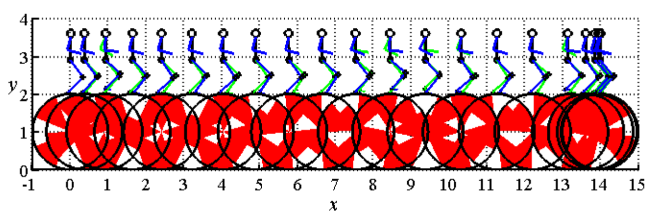


(a) (b) (c)

Fig. 9. Changing the desired θ_1^* from (a) to (b) and later to (c) at certain time frames while the robot is walking.

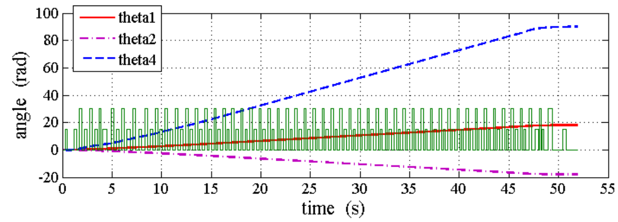


(a)

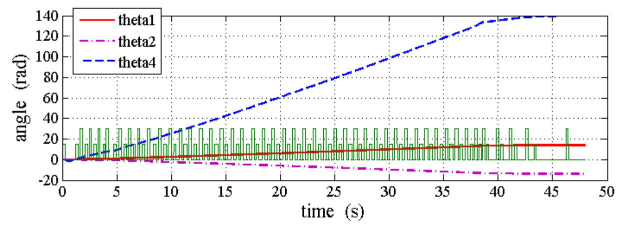


(b)

Fig. 10. Snapshots of the resulting motions to achieve the desired rolling speed $\dot{\theta}_1^r = 0.4$ rad/s. (a) The actual rolling speed of the ball of radius 0.5 m increases from $0 \rightarrow 0.2469 \rightarrow 0.3194 \rightarrow 0.3600 \rightarrow 0.3694 \rightarrow 0.4045$ rad/s. (b) The actual rolling speed of the unit ball increases from $0 \rightarrow 0.2318 \rightarrow 0.2943 \rightarrow 0.3414 \rightarrow 0.3685 \rightarrow 0.3983$ rad/s.

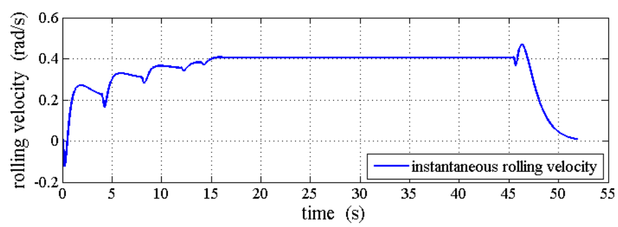


(a)

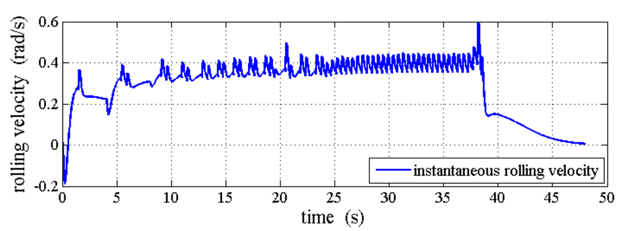


(b)

Fig. 11. Measured output of the simplified model. The desired rolling speed $\dot{\theta}_1^r = 0.4$ rad/s. (a) The rolling speed of the ball of radius 0.5 m increases from 0, through 0.2469, 0.3194, 0.3600, 0.3694, finally to 0.4045 rad/s. (b) The rolling speed of the unit ball increases from 0, through 0.2318, 0.2943, 0.3414, 0.3685, finally to 0.3983 rad/s.



(a)



(b)

Fig. 12. Instantaneous rolling velocity $\dot{\theta}_1$ of the ball. (a) $\dot{\theta}_1$ of the ball of radius 0.5 m stabilizes at the desired value. (b) $\dot{\theta}_1$ of the unit ball oscillates around the desired value due to the cyclic deviation of the actual CoP from the ideal CoP.

The snapshots are taken at a constant time interval. Thus it can be seen that the ball and the biped robot accelerate during the initial period and move at a constant speed afterwards. This result is also confirmed by the accompanying video as well as Fig. 11, which plots the measured θ_1 over the whole motion period.

Fig. 12 exhibits the instantaneous velocity $\dot{\theta}_1$ of the rolling ball over time. It is worth noticing that $\dot{\theta}_1$ of the ball of radius 0.5 m keeps stably close to the desired value $\dot{\theta}_1^r = 0.4$ rad/s, while $\dot{\theta}_1$ of the unit ball oscillates around $\dot{\theta}_1^r$. This is because the actual CoP from the footstep planner on the unit ball cyclically deviates from the ideal CoP generated by the balance controller, as revealed in Fig. 13.

VII. Conclusion and Future Work

In this paper, we present a control framework that enables a humanoid robot to walk on a rolling ball. It comprises a balance controller and a footstep planner. A wheeled linear inverted pendulum is used as a simplified model of the robot, for which a robust balance controller has been designed. The balance controller generates the ideal CoP as the input to the footstep planner, which computes the foot placements of the biped robot on the ball such that the actual CoP tracks the ideal CoP. Simulation results show that biped walking on balls of various sizes at different rolling speeds can be achieved by our method.

This work is just a beginning of the research on humanoid locomotion in non-stationary environments. There remain many possible directions to extend the work. First, we will design the controller for the frontal plane motion and generate motions in 3-D space. Second, we will explore better stepping strategies and more robust controllers and eventually realize this dynamic motion on a humanoid robot.

In hardware experiments, we need to consider how to measure the actual state on a real robot. One possible way is applying sufficient sensors to the system. Through the gyro sensor attached to the ball, we can detect the actual

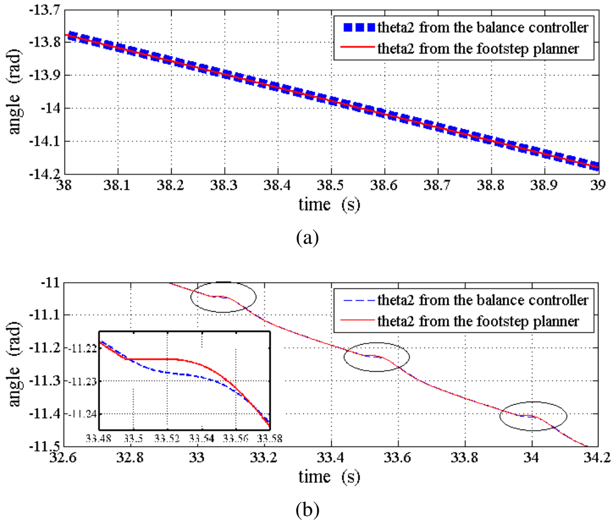


Fig. 13. Difference between the value of θ_2 from the balance controller and that from the footstep planner, which reflects the CoP deviation. (a) On the ball of radius 0.5 m, the actual CoP closely matches the ideal CoP. (b) On the unit ball, the actual CoP cyclically deviates from the ideal CoP. The deviation is shown in the enlarged view.

value of θ_1. Then the force-torque sensors at the ankles can provide us with the information to compute the actual CoP and θ_2. Finally the gyro sensor on the robot can help us determine the actual θ_4. We also need to consider sensor errors, which is why we have included an observer in our control framework. Even in the simulation, some of the angles may be discontinuous due to the discrete footsteps and simply taking their time derivatives would result in large velocities. The observer can act as a filter to prevent the system from overreacting to the discontinuous state variables.

REFERENCES

[1] Y. Kuroki, M. Fujita, T. Ishida, K. Nagasaka, and J. Yamaguchi, "A small biped entertainment robot exploring attractive applications," in *Proc. IEEE Int. Conf. Robot. Automat.*, Taipei, Taiwan, Sept. 2003, pp. 471–476.

[2] Y. Kuroki, K. Kato, K. Nagasaka, A. Miyamoto, K. Ueno, and J. Yamaguchi, "Motion evaluating system for a small biped entertainment robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, New Orieans, LA, Apr. 2004, pp. 3809–3814.

[3] S.-H. Hyon, "Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces," *IEEE Trans. Robot. Automat.*, vol. 25, no. 1, pp. 171–178, 2009.

[4] S. O. Anderson and J. K. Hodgins, "Methods for identifying and adapting models of humanoid robots performing dynamic tasks," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nashville, TN, Dec. 2010, pp. 511–517.

[5] S. Kajita and K. Tani, "Experimental study of biped dynamic walking in the linear inverted pendulum mode," in *Proc. IEEE Int. Conf. Robot. Automat.*, Nagoya, May 1995, pp. 2885–2891.

[6] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, Hawaii, Oct. 2001, pp. 239–246.

[7] 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. Automat.*, Washington, DC, May 2002, pp. 1404–1409.

[8] 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. Automat.*, Taipei, Taiwan, Sept. 2003, pp. 1620–1626.

[9] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Sys.*, Maui, HI, Oct. 2001, pp. 500–505.

[10] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep Planning for the Honda Asimo Humanoid," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, Apr. 2005, pp. 629–634.

[11] Y. Ayaz, T. Owa, T. Tsujita, A. Konno, K. Munawar, and M. Uchiyama, "Footstep planning for humanoid robots among obstacles of various types," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Paris, France, Dec. 2009, pp. 361–366.

[12] K. Yamane and J. Hodgins, "Control-aware mappping of human motion data with stepping for humanoid robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Taipei, Taiwan, Oct. 2010, pp. 726–733.

[13] K. Yamane and J. Hodgins, "Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, Oct. 2009, pp. 2510–2517.

[14] T. Sugihara, "Simulated regulator to synthesize ZMP manipulation and foot location for autonomous control of biped robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Pasadena, CA, May 2008, pp. 1264–1269.

[15] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: a step toward humanoid push recovery," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Paris, France, Dec. 2006, pp. 200–207.

[16] S. Kudoh, T. Komura, and K. Ikeuchi, "Stepping motion for a human-like character to maintain balance against large peturbations," in *Proc. IEEE Int. Conf. Robot. Automat.*, Orlando, Florida, May 2006, pp. 2661–2666.

[17] B. Stephens, "Humanoid push recovery," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Pittsburgh, PA, Dec. 2007, pp. 589–595.

[18] W. Mao, J.-J. Kim, and J.-J. Lee, " Continuous steps toward humanoid push recovery," in *Proc. IEEE Int. Conf. Automat. Logistics*, Shenyang, China, Aug. 2009, pp. 7–12.

[19] T. Endo and Y. Nakamura, "An omnidirectional vehicle on a basketball," in *Proc. Int. Conf. Advanced Robot.*, Seattle, WA, May 2005, pp.573–578.

[20] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proc. IEEE Int. Conf. Robot. Automat.*, Orlando, Florida, May. 2006, pp. 2884–2889.

[21] M. Kumagai and T. Ochiai, "Development of a robot balancing on a ball – Application of passive motion to transport," in *Proc. IEEE Int. Conf. Robot. Automat.*, Kobe, Japan, May 2009, pp. 4106–4111.