IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 January 2023, accepted 10 March 2023, date of publication 13 March 2023, date of current version 17 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3256720

RESEARCH ARTICLE

COM Shifter and Body Rotator for Step-by-Step Teleoperation of Bipedal Robots

YACHEN ZHANG AND RYO KIKUUWE, (Member, IEEE)
Machinery Dynamics Laboratory, Hiroshima University, Higashihiroshima, Hiroshima 739-8527, Japan
Corresponding author: Yachen Zhang (yachenzhang@mdl.hiroshima-u.ac.jp)

ABSTRACT This paper presents a controller for step-by-step teleoperation of bipedal robots, in which the user commands the robot's foot motions in a step-by-step manner through a pair of handheld 3-dimensional pointing devices. The proposed controller is intended to allow users to precisely manipulate the swing foot motions to traverse rough terrains by avoiding obstacles. It aims to realize quick response to user commands and stable automatic balancing even under erroneous user commands. The main components of the proposed controller are a COM (center-of-mass) shifter and a body rotator, which are simple sensory feedback controllers that do not involve first-in first-output (FIFO) buffers, time-series generation, or online optimization. The COM shifter is a controller to produce a COM motion according to a reference zero moment point (ZMP). The body rotator complements the COM shifter to produce an appropriate angular momentum rate to enhance the regulation of ZMP. The proposed controller is validated in our interactive/realtime simulation environment.

INDEX TERMS Teleoperation, bipedal robots, cart-flywheel-table model.

I. INTRODUCTION

The application of autonomous bipedal robots in hazardous environments is still limited by the current intelligence of robots. A possible solution to the difficulty is the use of teleoperation. For teleoperated bipedal robots, walking is one of the most important and basic tasks. Because a bipedal robot is an intrinsically unstable mechanism with many degrees of freedom (DOFs) and is prone to falling while walking, an appropriate combination of automatic balancing control and manual control is important for its teleoperation.

Schemes of teleoperated bipedal walking may be able to be classified into high-level and low-level teleoperation schemes. With the high-level teleoperation schemes, the user sends high-level commands, such as walking direction, walking speed, and target path, through some interface devices, such as joysticks [1], [2], [3] and graphical user interfaces (GUIs) [4], [5], [6], [7] on personal computers (PCs). Recently, devices that allow more intuitive operation, such as virtual reality (VR) interfaces [8] and foot-tapping walking-in-place interfaces [9], are also applied to send high-level commands. One concern with this type of schemes is that it requires automatic gain planning techniques (e.g., [10], [11], [12], [13], [14], [15]) to generate stable walking patterns, and it is usually computationally costly. In addition, because there will always be limits to the reliability of automatic planning techniques in complex or unpredictable terrains, it would be necessary to prepare options to allow users to operate the robot in a more direct manner.

Low-level teleoperation schemes, in which the user directly commands the body and leg motions of the robot, can be a useful option to cope with cases where automatic gait planning techniques are unavailable or inapplicable. This type of schemes requires some means to receive the information of the user's body motion to be mapped to the robot's body motion. Many of low-level teleoperation schemes so far employ complex mechatronic devices that restrain the user's body, such as motion capture systems [16], [17], exoskeletons [18], [19], [20], and whole-body interfaces [21], [22], [23], [24]. Even with sophisticated whole-body interface devices nevertheless, practical teleoperation is not straightforward because the exact position matching of body parts between the user and the robot may cause the loss of balance of the robot. Matching the center of mass (COM) and the

The associate editor coordinating the review of this manuscript and approving it for publication was Mouloud Denai.

zero moment point (ZMP) between the user and the robot enables dynamic and intuitive teleoperation [22], but it is not very suited for traversing rough terrain by carefully choosing every footstep, avoiding the collision of the swing foot and obstacles. In such cases, not only the footstep positions but also the swing-foot trajectory need to be manipulated by the user to avoid the collision between the foot and the environment. As far as the authors are aware, there have been no such methods for teleoperated bipedal walking that allow precise manipulation of the swing foot in the single support phase.

This paper proposes a controller for intuitively teleoperating a bipedal robot through a pair of inexpensive, handheld 3-dimensional (3D) pointing devices, as in Fig. 1. The two pointing devices correspond to the two feet of the robot. The controller allows for what we call a step-by-step teleoperation scheme, in which the user manipulates the foot motion at every step of walking. In the double support phase, lifting one pointing device leads to its corresponding foot being lifted. In the single support phase, the user commands the swing-foot position relative to the support-foot position through the corresponding pointing device. It allows the user to make the robot walk across obstacles in complex environments by carefully moving the swing foot and choosing suitable footholds. The scheme is designed to allow the user to teleoperate the robot without force feedback, which would cause stability problems especially when there is a communication delay between the devices and the robot.

The step-by-step teleoperation scheme can be classified as a low-level teleoperation scheme because the automatic gait planning is not used while walking. Advantages of this scheme over other low-level teleoperation schemes, e.g., [16], [17], [18], [19], [20], [21], [22], [23], and [24], can be said to be relatively low hardware cost and relatively low physical burden to the user. Using the hands instead of legs would be convenient for carefully operating the robot's feet especially when the user does not prefer to be restrained by complicated devices. The scheme provides better maneuverability of the swing foot for traversing obstacles, although it may not be efficient in flat or structured environments. A potentially useful scenario is that, when the environment is found to be too complex for the robot's autonomous planning function, this scheme could be activated to entrust the motion planning to the human user's decisions. According to the classification presented in Table 1 of the recent survey paper [25], the presented scheme can be said to be a method with a joystick interface, without any haptic feedback, with a "footstep motion generation," with a ZMP stabilizer, and with joint position control. It may be worth mentioning that Table 1 in [25] does not refer to such work from the literature.

Early versions of the proposed controller have been presented in previous conference publications [26], [27] from our research group. This paper presents the overall refinement of the controller architecture with new main components, a COM shifter and a body rotator, which are combined through prioritized differential inverse kinematics (PDIK) with a carefully designed strategy to determine the weight matrices. The whole controller is a simple sensory feedback controller that is free from first-in first-output (FIFO) buffers, time-series generators, or online optimization. The COM shifter, one of the main components, can be seen as an online ZMP-based walking pattern generator [28, Section 4.4.3], but is much simpler than those with FIFO buffers [29]. The body rotator, the other one, enhances the stability by adjusting the angular momentum rate of the body, especially for suppressing the effect of the swing-leg motions caused by rough commands from the user. Using arm motions may be an option to generate the reaction torque as has been the case in, e.g., [30] and [31], but we only use the body rotation to leave the arms available for other tasks in practical situations. The proposed controller is validated in our interactive/realtime simulation environment shown in Fig. 1.

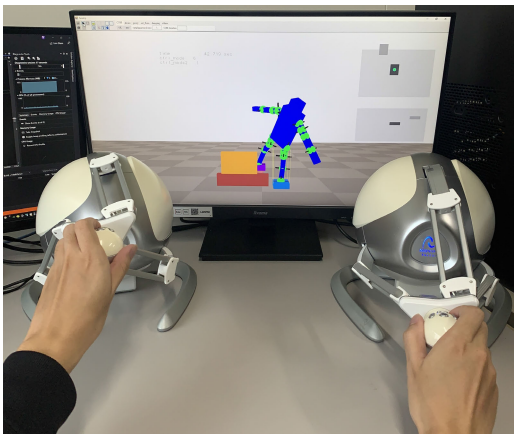

FIGURE 1. Interactive simulation setup for the step-by-step teleoperation of a bipedal robot.

This paper is organized as follows. Section II gives some preliminaries. Section III introduces the overall architecture of our controller. Section IV details the main components of our controller; the COM shifter and the body rotator. Section V shows the results of some simulations, including some comparative results. Conclusions are provided in Section VI.

II. PRELIMINARIES

A. COORDINATE FRAMES

We consider a bipedal robot as a floating-base system composed of $6 + n$ DOFs, as shown in Fig. 2, where n is the number of joints of the robot. Each leg has 6 DOFs. There are four coordinate frames used in our framework, which are Σ_W, Σ_B, Σ_L and Σ_R. Here, Σ_W is the world coordinate frame, Σ_B is the coordinate frame fixed to the torso link, and Σ_L and Σ_R are the coordinate frames fixed to the left foot and right foot, respectively. The joint angle vector of the robot is denoted by $\boldsymbol{q}_A \in \mathbb{R}^n$. Throughout this paper, vectors with subscripts B, L, and R are associated with the correspondent coordinate frames. The subscript G corresponds to the COM of the robot. The subscript S corresponds to the swing foot in the single support phase and the right foot in the double support phase. In this paper, unless otherwise specified, all

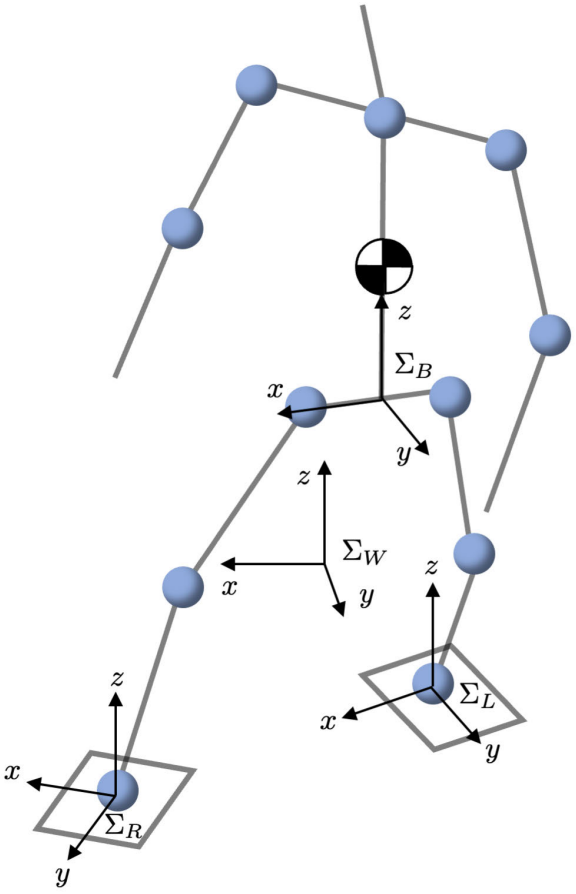

FIGURE 2. Coordinate frames associated with a bipedal robot.

vectors of position, velocity, angular velocity, and momentum are represented in the world coordinate system Σ_W.

B. ZMP EQUATION

With a robot of which at least one of its feet is grounded as in Fig. 2, the relation among the ZMP $\boldsymbol{r} \in \mathbb{R}^3$, the COM $\boldsymbol{p}_G \in \mathbb{R}^3$, and the angular momentum $\boldsymbol{L}_G \in \mathbb{R}^3$ about the COM can be given as follows [13], [32], [33]:

$$\begin{cases} r_x = \dfrac{(g+\ddot{p}_{Gz})p_{Gx} - (p_{Gz}-r_z)\ddot{p}_{Gx} - \dot{L}_{Gy}/m}{g+\ddot{p}_{Gz}} \\ r_y = \dfrac{(g+\ddot{p}_{Gz})p_{Gy} - (p_{Gz}-r_z)\ddot{p}_{Gy} + \dot{L}_{Gx}/m}{g+\ddot{p}_{Gz}}. \end{cases} \tag{1}$$

Here, the subscripts x, y, and z stand for the x, y, and z components, respectively, of the associated vectors, m is the total mass of the robot, and g is the gravitational acceleration.

Because ZMP is always on the ground, one can set $r_z = 0$. In addition, if the vertical motion of COM is negligible, we can assume that $\ddot{p}_{Gz} = 0$. Then, the ZMP equation under such an assumption is obtained as follows:

$$\begin{cases} r_x = p_{Gx} - \dfrac{\ddot{p}_{Gx}}{g/p_{Gz}} - \dfrac{\dot{L}_{Gy}}{mg} \\ r_y = p_{Gy} - \dfrac{\ddot{p}_{Gy}}{g/p_{Gz}} + \dfrac{\dot{L}_{Gx}}{mg}. \end{cases} \tag{2}$$

This formulation corresponds to the linear inverted pendulum (LIP) plus flywheel model [34] if $\boldsymbol{r}$ is seen as an input. If the last terms involving $\dot{\boldsymbol{L}}_G$ are neglected, it reduces to the LIP model. Moreover, if $\ddot{\boldsymbol{p}}_{Gxy}$ is seen as the input and $\dot{\boldsymbol{L}}_G$ is neglected, it can be seen as the cart-table model [29]. Some comprehensive discussions on the structure (2), involving $\dot{\boldsymbol{L}}_G$, are found in [33].

The majority of the previous techniques, e.g., [11], [12], [14], [27], [29], [35], [36], and [37], are built on the reduced model without the $\dot{\boldsymbol{L}}_G$ terms, i.e., the LIP model or the cart-table model. There have also been many controllers accounting for the $\dot{\boldsymbol{L}}_G$ terms [13], [30], [32], [38], many of which depend on predetermined motion commands. This paper builds a simple controller based on the full model (2) with $\ddot{\boldsymbol{p}}_{Gxy}$ and $\dot{\boldsymbol{L}}_{Gxy}$ treated as inputs, which we call a *cart-flywheel-table model*, as detailed in Section IV.

C. COM VELOCITY AND ANGULAR MOMENTUM

Let $\boldsymbol{p}_B \in \mathbb{R}^3$ and $\boldsymbol{\omega}_B \in \mathbb{R}^3$ be the position and the angular velocity of Σ_B, respectively. Then, the velocity $\dot{\boldsymbol{p}}_G$ of COM and the angular momentum $\boldsymbol{L}_G$ about the COM can be obtained as follows:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_G \\ \boldsymbol{L}_G \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & -[\boldsymbol{p}_{GB}\times] & \hat{\boldsymbol{M}}_G \\ \boldsymbol{0} & \tilde{\boldsymbol{H}} & \boldsymbol{H}_G \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_B \\ \boldsymbol{\omega}_B \\ \dot{\boldsymbol{q}}_A \end{bmatrix} \tag{3}$$

where

$$\hat{\boldsymbol{M}}_G \triangleq \boldsymbol{M}_G/m. \tag{4}$$

Here, $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix, $\boldsymbol{p}_{GB} \in \mathbb{R}^3$ is the position vector of the robot's COM from the origin of Σ_B, $\tilde{\boldsymbol{H}} \in \mathbb{R}^{3\times3}$ is the total moment of inertia of the robot about the COM, $\boldsymbol{M}_G \in \mathbb{R}^{3\times n}$ and $\boldsymbol{H}_G \in \mathbb{R}^{3\times n}$ are the inertia matrices that relate the joint velocities into the linear momentum and the angular momentum of the robot, respectively, and $[()\times]$ is the operator that translates a 3-vector into a 3×3 skew symmetric matrix equivalent to the cross product. The matrices $\tilde{\boldsymbol{H}}$, $\boldsymbol{M}_G$ and $\boldsymbol{H}_G$ can be calculated in realtime through an efficient computation method, such as the one proposed in [39].

When the robot is floating in the air, $\dot{\boldsymbol{p}}_G$ and $\boldsymbol{L}_G$ are expressed by (3). In general, the DOF of the robot is reduced due to the contact with the ground. To obtain the constrained form of (3), we divide $\dot{\boldsymbol{q}}_A$, $\hat{\boldsymbol{M}}_G$ and $\boldsymbol{H}_G$ into leg parts and the other part in the following manner:

$$\dot{\boldsymbol{q}}_A = [\dot{\boldsymbol{q}}_L^T, \dot{\boldsymbol{q}}_R^T, \dot{\boldsymbol{q}}_o^T]^T \tag{5}$$

$$\hat{\boldsymbol{M}}_G = [\hat{\boldsymbol{M}}_L, \hat{\boldsymbol{M}}_R, \hat{\boldsymbol{M}}_o] \tag{6}$$

$$\boldsymbol{H}_G = [\boldsymbol{H}_L, \boldsymbol{H}_R, \boldsymbol{H}_o]. \tag{7}$$

Here, $\dot{\boldsymbol{q}}_L \in \mathbb{R}^6$, $\hat{\boldsymbol{M}}_L \in \mathbb{R}^{3\times6}$, and $\boldsymbol{H}_L \in \mathbb{R}^{3\times6}$ correspond to the left leg, $\dot{\boldsymbol{q}}_R \in \mathbb{R}^6$, $\hat{\boldsymbol{M}}_R \in \mathbb{R}^{3\times6}$, and $\boldsymbol{H}_R \in \mathbb{R}^{3\times6}$ correspond to the right leg, and $\dot{\boldsymbol{q}}_o \in \mathbb{R}^6$, $\hat{\boldsymbol{M}}_o \in \mathbb{R}^{3\times6}$, and $\boldsymbol{H}_o \in \mathbb{R}^{3\times6}$ correspond to the rest of the robot (i.e., the body

and the arms). Then, (3) can be rewritten as follows:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_G \\ \boldsymbol{L}_G \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & -[\boldsymbol{p}_{GB}\times] \\ \boldsymbol{0} & \tilde{\boldsymbol{H}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_B \\ \boldsymbol{\omega}_B \end{bmatrix} + \begin{bmatrix} \hat{\boldsymbol{M}}_L \\ \boldsymbol{H}_L \end{bmatrix} \dot{\boldsymbol{q}}_L + \begin{bmatrix} \hat{\boldsymbol{M}}_R \\ \boldsymbol{H}_R \end{bmatrix} \dot{\boldsymbol{q}}_R + \begin{bmatrix} \hat{\boldsymbol{M}}_o \\ \boldsymbol{H}_o \end{bmatrix} \dot{\boldsymbol{q}}_o. \quad (8)$$

Let $\boldsymbol{p}_L \in \mathbb{R}^3$ and $\boldsymbol{\omega}_L \in \mathbb{R}^3$ be the position and the angular velocity, respectively, of Σ_L, and let $\boldsymbol{p}_R \in \mathbb{R}^3$ and $\boldsymbol{\omega}_R \in \mathbb{R}^3$ be the position and the angular velocity, respectively, of Σ_R. They are obtained by the following expression:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_* \\ \boldsymbol{\omega}_* \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & [\boldsymbol{p}_{*B}\times] \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{p}}_B \\ \boldsymbol{\omega}_B \end{bmatrix} + \boldsymbol{J}_* \dot{\boldsymbol{q}}_* \quad (9)$$

where the subscript $*$ can be L or R corresponding to the left or right foot, respectively, $\boldsymbol{p}_{*B} \in \mathbb{R}^3$ is the position vector of the origin of the foot coordinate Σ_* seen in Σ_B, and $\boldsymbol{J}_* \in \mathbb{R}^{6\times n}$ is the Jacobian matrix calculated from the corresponding leg configuration.

When the left foot is grounded, its velocity is constrained as follows:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_L \\ \boldsymbol{\omega}_L \end{bmatrix} = \boldsymbol{0}. \quad (10)$$

By substituting (10) into (9), $\dot{\boldsymbol{p}}_B$ and $\boldsymbol{\omega}_B$ can be given by

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{p}}_B \\ \boldsymbol{\omega}_B \end{bmatrix} &= -\begin{bmatrix} \boldsymbol{I} & [\boldsymbol{p}_{LB}\times] \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \boldsymbol{J}_L \dot{\boldsymbol{q}}_L \\ &= -\begin{bmatrix} \boldsymbol{I} & -[\boldsymbol{p}_{LB}\times] \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{J}_L \dot{\boldsymbol{q}}_L. \end{aligned} \quad (11)$$

Hence, we obtain $\dot{\boldsymbol{p}}_G$ and $\boldsymbol{L}_G$ of the robot under the constraint (10) by substituting (11) into (8) as follows:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_G \\ \boldsymbol{L}_G \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{M}}^*_G \\ \boldsymbol{H}^*_G \end{bmatrix} \dot{\boldsymbol{q}}_A \quad (12)$$

where

$$\begin{bmatrix} \hat{\boldsymbol{M}}^*_G \\ \boldsymbol{H}^*_G \end{bmatrix} \triangleq \begin{bmatrix} \hat{\boldsymbol{M}}^*_B + \hat{\boldsymbol{M}}_L & \hat{\boldsymbol{M}}_R & \hat{\boldsymbol{M}}_o \\ \boldsymbol{H}^*_B + \boldsymbol{H}_L & \boldsymbol{H}_R & \boldsymbol{H}_o \end{bmatrix} \quad (13a)$$

$$\begin{aligned} \begin{bmatrix} \hat{\boldsymbol{M}}^*_B \\ \boldsymbol{H}^*_B \end{bmatrix} &\triangleq -\begin{bmatrix} \boldsymbol{I} & -[\boldsymbol{p}_{GB}\times] \\ \boldsymbol{0} & q\tilde{\boldsymbol{H}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -[\boldsymbol{p}_{LB}\times] \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{J}_L \\ &= -\begin{bmatrix} \boldsymbol{I} & -[(\boldsymbol{p}_{GB}+\boldsymbol{p}_{LB})\times] \\ \boldsymbol{0} & \tilde{\boldsymbol{H}} \end{bmatrix} \boldsymbol{J}_L. \end{aligned} \quad (13b)$$

The case where the right foot is grounded is described by (12) with the subscripts L and R being interchanged in (13). Please notice that $\hat{\boldsymbol{M}}^*_G$ and $\boldsymbol{H}^*_G$ are Jacobian-like matrices that transform the joint velocities to $\dot{\boldsymbol{p}}_G$ and $\boldsymbol{L}_G$ in the task space, although they depend on mass/inertia parameters.

III. OVERALL STRUCTURE OF THE CONTROLLER

A. FRAMEWORK

This section presents our controller framework for step-by-step teleoperation of bipedal walking. It is illustrated in Fig. 3. Early versions of the framework have been introduced in our prior conference publications [26], [27], but the present version is now much simplified in some aspects, mainly by pruning unnecessary components and modes. It now comprises two new components, a COM shifter and a body rotator, which will be detailed in the next Section IV, and they are combined through PDIK with carefully designed weight matrices. This section focuses on the framework into which these new components are incorporated.

In this teleoperation scheme, the robot's feet are manipulated in realtime by the user. More precisely, as shown in Fig. 1, there are two 3D pointing devices, the left-hand and the right-hand pointing devices, which manipulate the left foot and the right foot, respectively. In the double support phase, lifting one pointing device leads to its corresponding foot being lifted. In the single support phase, the swing foot follows the motion of its correspondent pointing device until the sensors detect its contact with the ground. The position commands from the pointing devices appear in Fig. 3 as $\boldsymbol{p}_L^{\text{device}} \in \mathbb{R}^3$ and $\boldsymbol{p}_R^{\text{device}} \in \mathbb{R}^3$, which are the inputs to the reference generator.

Now, the structure of the framework is explained from left to right in Fig. 3. The reference generator, the left-most block, accepts the position commands from the pointing devices and foot positions $\boldsymbol{p}_L$ and $\boldsymbol{p}_R$ from the forward kinematic computations, and generates reference values of ZMP and position/attitude information. Its details are presented in Section III-C. The reference values are translated into desired velocity information represented by a vector $\boldsymbol{v}_d \in \mathbb{R}^{14}$. This process is performed by the COM shifter, the body rotator, and the saturated proportional (P) controller. The former two, the main components, are detailed in Section IV and the last one is detailed in Section III-C. The PDIK block in the middle of Fig. 3 is to convert the desired velocity vector $\boldsymbol{v}_d \in \mathbb{R}^{14}$ into the angular velocity command $\boldsymbol{u}_d \in \mathbb{R}^{12}$, which is sent to 12 joints of the two legs. This PDIK block will be detailed in Section III-D. In the computation of PDIK, the joint velocity commands $\boldsymbol{u}_d$ are determined so that the joint angles do not exceed the motion range limits, as detailed in Section III-D. The whole controller is governed by five different modes illustrated in Fig. 4 according to walking phases, as will be detailed in Section III-B.

As can be seen in Fig. 3, the output $\boldsymbol{u}_d$ of PDIK is sent to angular velocity controllers of the joints, each of which is assumed to be velocity-controlled by an independent servo controller. We only consider the joints of two legs, of which the joint angles are aggregated into the joint angle vector $\boldsymbol{q} \triangleq [\boldsymbol{q}_L^T, \boldsymbol{q}_R^T]^T \in \mathbb{R}^{12}$, and other joints are set to constant angles.

B. MODE TRANSITIONS

The mode transition diagram of the controller is shown in Fig. 4. It is similar to the one in our prior work [27], but there have been some modifications. In Fig. 4, the mode D is the double-support mode, the modes SL and SR are the modes of single support by the left foot and by the right foot, respectively, and the modes TL and TR are transient modes to the left and right single-support modes, respectively.

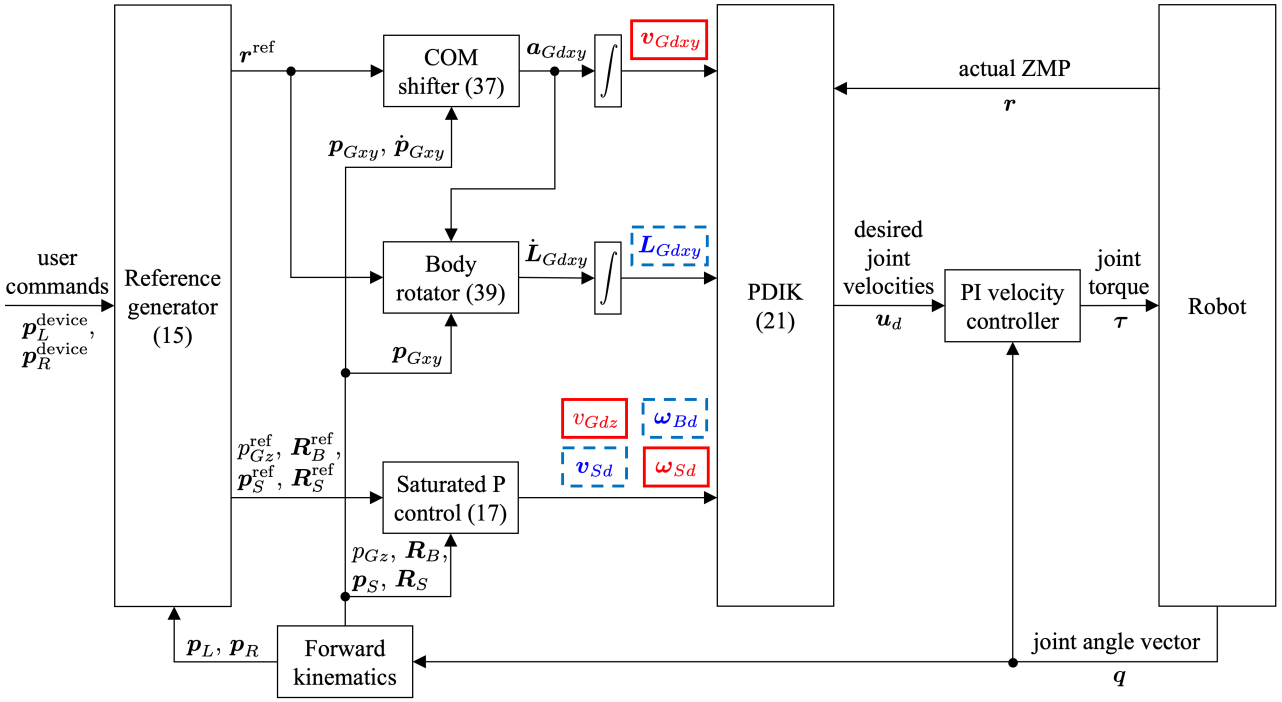

FIGURE 3. Block diagram of the proposed controller. The components of the desired velocity vector $\nu_d \in \mathbb{R}^{14}$ are indicated by red solid boxes and blue dashed boxes, which correspond to high- and low-priority components, respectively.

The trigger events e_* in the figure are defined as follows:

$$e_{\text{DTL}} : \text{the right pointing device lifted} \tag{14a}$$
$$e_{\text{DTR}} : \text{the left pointing device lifted} \tag{14b}$$
$$e_{\text{SDL}} : \text{the left foot grounded} \tag{14c}$$
$$e_{\text{SDR}} : \text{the right foot grounded} \tag{14d}$$
$$e_{\text{TS}} : \|\boldsymbol{r} - \boldsymbol{r}^{\text{ref}}\| \leq r_{\text{lift}}. \tag{14e}$$

Here, the actual ZMP $\boldsymbol{r}$ is measured by load cells at each foot sole, and $\boldsymbol{r}^{\text{ref}} \in \mathbb{R}^3$ is the reference ZMP determined by the reference generator introduced in Section III-C. The events e_{DTL} and e_{DTR} are triggered when the z components of the pointing device positions $\boldsymbol{p}_L^{\text{device}}$ and $\boldsymbol{p}_R^{\text{device}}$ become positive, respectively, and the events e_{SDL} and e_{SDR} are detected through load cells. The controller parameter r_{lift} should be set according to the robot foot size to ensure that $\boldsymbol{r}$ is within the support polygon when one foot is lifted. We set r_{lift} as 0.12 m in the human-sized robot used in the simulator reported in Section V.

In the mode D, the robot is supposed to be in the double support phase, and the COM is controlled to converge to the above of the midpoint of the feet until the event e_{DTL} or e_{DTR} is created by the user.

In the modes SL and SR, the robot is supposed to be in the single support phase, and the COM is controlled to the above of the support foot until the event e_{SDL} or e_{SDR} is created by the user's operation to lower the swing foot to the ground. Only in these two modes, the user is allowed to manipulate the swing foot in realtime.

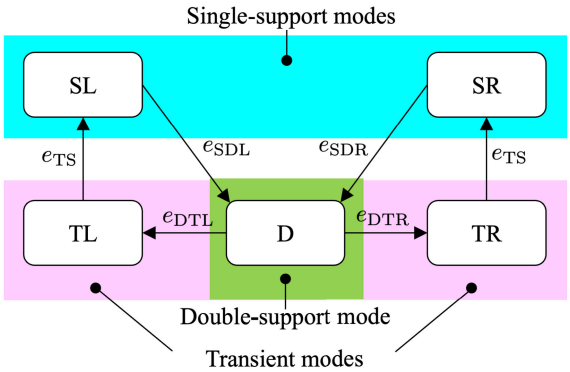

FIGURE 4. Mode transition diagram of the proposed controller.

In the transient modes TL and TR, the ZMP is shifted to the corresponding support foot until the event e_{TS} in (14e) takes place. After that, the single-support mode SL or SR is initiated and the user is allowed to lift the foot.

C. REFERENCE GENERATOR & DESIRED VELOCITY VECTOR

The reference generator sends the following five quantities:

- $\boldsymbol{r}^{\text{ref}} \in \mathbb{R}^3$: ZMP's reference position,
- $p_{Gz}^{\text{ref}} \in \mathbb{R}$: z component of the COM's reference position,
- $\boldsymbol{R}_B^{\text{ref}} \in \mathbb{R}^{3\times3}$: torso's reference attitude,
- $\boldsymbol{p}_S^{\text{ref}} \in \mathbb{R}^3$: swing foot's reference position, and
- $\boldsymbol{R}_S^{\text{ref}} \in \mathbb{R}^{3\times3}$: swing foot's reference attitude.

Recall that the center positions of the left and right foot soles are denoted by $\boldsymbol{p}_L$ and $\boldsymbol{p}_R$, respectively. To maintain the balance of the robot, in the double support phase, ZMP should be in the middle of the two feet, and when a foot is commanded to be lifted, ZMP should be moved to the other foot, which will become the support foot. Based on this idea, the reference generator determines the above quantities as follows:

$$\boldsymbol{r}^{\mathrm{ref}} = \begin{cases} (\boldsymbol{p}_L + \boldsymbol{p}_R)/2 & \text{if D} \\ \boldsymbol{p}_L & \text{if TL} \vee \text{SL} \\ \boldsymbol{p}_R & \text{if TR} \vee \text{SR} \end{cases} \tag{15a}$$

$$p_{Gz}^{\mathrm{ref}} = H_G \tag{15b}$$

$$\boldsymbol{R}_B^{\mathrm{ref}} = \boldsymbol{I} \tag{15c}$$

$$\boldsymbol{p}_S^{\mathrm{ref}} = \begin{cases} \boldsymbol{p}_R^{\mathrm{device}} & \text{if SL} \\ \boldsymbol{p}_L^{\mathrm{device}} & \text{if SR} \\ \boldsymbol{p}_R & \text{if TL} \vee \text{D} \\ \boldsymbol{p}_L & \text{if TR} \end{cases} \tag{15d}$$

$$\boldsymbol{R}_S^{\mathrm{ref}} = \boldsymbol{I}. \tag{15e}$$

Here, H_G is a constant representing the nominal height of the COM, which was set as 0.765 m in the human-sized robot used in the simulator reported in Section V. Note that, in (15a) and (15d), $\boldsymbol{r}^{\mathrm{ref}}$ and $\boldsymbol{p}_S^{\mathrm{ref}}$ are defined for all five modes, and thus their values are determined at any time instants.

As shown in Fig. 3, the generated reference values are converted into the following desired velocity values:

- $\boldsymbol{v}_{Gdxy} \in \mathbb{R}^2$: x and y components of the COM's translational velocity,
- $\boldsymbol{L}_{Gdxy} \in \mathbb{R}^2$: x and y components of the angular momentum about COM,
- $v_{Gdz} \in \mathbb{R}$: z component of the COM's translational velocity,
- $\boldsymbol{\omega}_{Bd} \in \mathbb{R}^3$: torso's angular velocity,
- $\boldsymbol{v}_{Sd} \in \mathbb{R}^3$: swing foot's translational velocity, and
- $\boldsymbol{\omega}_{Sd} \in \mathbb{R}^3$: swing foot's angular velocity.

These values are aggregated into the following desired velocity vector:

$$\boldsymbol{v}_d \triangleq [\boldsymbol{v}_{Gdxy}^T, \boldsymbol{L}_{Gdxy}^T, v_{Gdz}, \boldsymbol{\omega}_{Bd}^T, \boldsymbol{v}_{Sd}^T, \boldsymbol{\omega}_{Sd}^T]^T \in \mathbb{R}^{14}. \tag{16}$$

Among these values, $\boldsymbol{v}_{Gdxy}$ and $\boldsymbol{L}_{Gdxy}$ are determined by the reference ZMP $\boldsymbol{r}^{\mathrm{ref}}$ through the COM shifter and the body rotator as will be detailed in Section IV.

The rest 10 components of $\boldsymbol{v}_d$ are determined by simple saturated proportional (P) controllers. Let p_{Gz} be the z components of the actual COM $\boldsymbol{p}_G$, $\boldsymbol{p}_S$ be the actual position of the swing foot, and $\boldsymbol{R}_* \in \mathbb{R}^{3\times3}$ ($* \in \{S, B\}$) be the rotation matrices representing the actual attitudes of the frames of Σ_*. Then, the saturated P controllers to determine the desired velocity values are written as follows:

$$v_{Gdz} = \mathrm{sat}(v_{\mathrm{lim}}, k_v(p_{Gz}^{\mathrm{ref}} - p_{Gz})) \tag{17a}$$

$$\boldsymbol{\omega}_{Bd} = \mathrm{sat}(\omega_{\mathrm{lim}}, k_\omega(\ln \boldsymbol{R}_B^{\mathrm{ref}} \boldsymbol{R}_B^T)^\vee) \tag{17b}$$

$$\boldsymbol{v}_{Sd} = \mathrm{sat}(v_{\mathrm{lim}}, k_v(\boldsymbol{p}_S^{\mathrm{ref}} - \boldsymbol{p}_S)) \tag{17c}$$

$$\boldsymbol{\omega}_{Sd} = \mathrm{sat}(\omega_{\mathrm{lim}}, k_\omega(\ln \boldsymbol{R}_S^{\mathrm{ref}} \boldsymbol{R}_S^T)^\vee) \tag{17d}$$

where $\mathrm{sat} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$\mathrm{sat}(x_{\mathrm{lim}}, \boldsymbol{x}) \triangleq \frac{x_{\mathrm{lim}} \boldsymbol{x}}{\max(x_{\mathrm{lim}}, \|\boldsymbol{x}\|)}, \tag{18}$$

and the notation $(\ln \boldsymbol{R}_a \boldsymbol{R}_b^T)^\vee$ represents the rotation vector from the attitude $\boldsymbol{R}_b$ to the attitude $\boldsymbol{R}_a$, which is detailed in Appendix.

The velocity limits v_{lim} and ω_{lim} can be chosen based on the capacities of the joint actuators of the robot. The gains k_v and k_ω can be chosen according to how fast the convergence should be, considering that the gains can be interpreted as the inverses of the time constants of the exponential convergence. They were set as $\{v_{\mathrm{lim}}, \omega_{\mathrm{lim}}, k_v, k_\omega\} = \{1\ \mathrm{m/s}, 0.8\ \mathrm{rad/s}, 10\ \mathrm{s}^{-1}, 10\ \mathrm{s}^{-1}\}$ for the human-sized robot used in Section V.

D. PRIORITIZED DIFFERENTIAL INVERSE KINEMATICS (PDIK)

Since there are only 12 DOFs in two legs of the robot, $\boldsymbol{v}_d \in \mathbb{R}^{14}$ cannot be realized completely. Furthermore, when the robot reaches the motion range limits of the joints or the singular configurations, it results in the reduction of DOFs. To avoid this problem, we define thresholds $\boldsymbol{q}_{\max} \in \mathbb{R}^{12}$ and $\boldsymbol{q}_{\min} \in \mathbb{R}^{12}$ of legs' joint angles, which ensure that $\boldsymbol{q} \in \{\boldsymbol{x} \in \mathbb{R}^{12} \mid \boldsymbol{q}_{\min} \le \boldsymbol{x} \le \boldsymbol{q}_{\max}\}$ are within the motion range limits and are not in singular configurations. To obtain an appropriate set of velocity angle commands $\boldsymbol{u}_d \in \mathbb{R}^{12}$, we employ the PDIK based on the method proposed in [40].

We divide the desired velocity vector $\boldsymbol{v}_d$ in (16) into the following two parts:

$$\boldsymbol{v}_{d1} \triangleq [\boldsymbol{v}_{Gdxy}^T, v_{Gdz}, \boldsymbol{\omega}_{Sd}^T]^T \in \mathbb{R}^6 \tag{19a}$$

$$\boldsymbol{v}_{d2} \triangleq [\boldsymbol{L}_{Gdxy}^T, \boldsymbol{\omega}_{Bd}^T, \boldsymbol{v}_{Sd}^T]^T \in \mathbb{R}^8. \tag{19b}$$

Here, $\boldsymbol{v}_{d1}$ and $\boldsymbol{v}_{d2}$ are the high- and low-priority desired velocities, respectively. This classification has been determined by considering how accurately the elements should be realized.

- The COM's translational velocities $\boldsymbol{v}_{Gdxy}$ and v_{Gdz} are crucial for the stability of motion. Thus, they must be realized with high accuracy all the time.
- The swing foot's angular velocity $\boldsymbol{\omega}_{Sd}$ should always be realized accurately because otherwise the foot may be grounded in an unexpected posture.
- The angular momentum $\boldsymbol{L}_{Gdxy}$ is allowed to be relatively inaccurate due to its relatively small effect on the balance in the double support phase.
- The remained two elements $\boldsymbol{\omega}_{Bd}$ and $\boldsymbol{v}_{Sd}$ should not be realized accurately when they cause errors in realization of angular momentum in the single support phase. Otherwise, the robot will lose balance.

The joint velocity command $\boldsymbol{u}_d$ is generated from $\boldsymbol{v}_{d1}$ and $\boldsymbol{v}_{d2}$ by solving the following constrained quadratic optimization problem:

$$\min_{\boldsymbol{u}_d} \ \|\boldsymbol{J}_2\boldsymbol{u}_d - \boldsymbol{v}_{d2}\|^2_{\boldsymbol{W}_A} + \|\boldsymbol{u}_d\|^2_{\boldsymbol{W}_B}$$
$$\text{s.t. } \boldsymbol{J}_1\boldsymbol{u}_d = \boldsymbol{v}_{d1} \tag{20}$$

where $\boldsymbol{J}_1 \in \mathbb{R}^{6\times12}$ and $\boldsymbol{J}_2 \in \mathbb{R}^{8\times12}$ are the Jacobian matrices that relate $\boldsymbol{u}_d$ to $\boldsymbol{v}_{d1}$ and $\boldsymbol{v}_{d2}$, respectively, the notation $\|\boldsymbol{z}\|_{\boldsymbol{W}}$ stands for $\|\boldsymbol{z}\|_{\boldsymbol{W}} = \sqrt{\boldsymbol{z}^T\boldsymbol{W}\boldsymbol{z}}$, which is the norm of $\boldsymbol{z}$ with the metric matrix $\boldsymbol{W}$, and $\boldsymbol{W}_A \in \mathbb{R}^{8\times8}$ and $\boldsymbol{W}_B \in \mathbb{R}^{12\times12}$ are diagonal and positive definite matrices to be designed. The solution of the optimization problem (20) is analytically obtained as follows:

$$\boldsymbol{u}_d = \boldsymbol{W}_B^{-1/2}\boldsymbol{J}_{1W}^{+}\boldsymbol{v}_{d1} + \boldsymbol{W}_B^{-1/2}\tilde{\boldsymbol{J}}_{2W}^{\#}(\boldsymbol{v}_{d2} - \boldsymbol{J}_{2W}\boldsymbol{J}_{1W}^{+}\boldsymbol{v}_{d1}) \tag{21}$$

where

$$\boldsymbol{J}_{1W} \triangleq \boldsymbol{J}_1\boldsymbol{W}_B^{-1/2} \in \mathbb{R}^{6\times12} \tag{22a}$$
$$\boldsymbol{J}_{2W} \triangleq \boldsymbol{J}_2\boldsymbol{W}_B^{-1/2} \in \mathbb{R}^{8\times12} \tag{22b}$$
$$\boldsymbol{J}_{1W}^{+} \triangleq \boldsymbol{J}_{1W}^T(\boldsymbol{J}_{1W}\boldsymbol{J}_{1W}^T)^{-1} \in \mathbb{R}^{12\times6} \tag{22c}$$
$$\tilde{\boldsymbol{J}}_{2W} \triangleq \boldsymbol{J}_{2W}(\boldsymbol{I} - \boldsymbol{J}_{1W}^{+}\boldsymbol{J}_{1W}) \in \mathbb{R}^{8\times12} \tag{22d}$$
$$\tilde{\boldsymbol{J}}_{2W}^{\#} \triangleq (\tilde{\boldsymbol{J}}_{2W}^T\boldsymbol{W}_A\tilde{\boldsymbol{J}}_{2W} + \boldsymbol{I})^{-1}\tilde{\boldsymbol{J}}_{2W}^T\boldsymbol{W}_A \in \mathbb{R}^{12\times8}. \tag{22e}$$

Here, $\boldsymbol{J}_{1W}^{+}$ is said to be the right inverse of $\boldsymbol{J}_{1W}$ and $\tilde{\boldsymbol{J}}_{2W}^{\#}$ is said to be a singularity robust inverse [41] of $\tilde{\boldsymbol{J}}_{2W}$.

The design of the diagonal matrix $\boldsymbol{W}_A$ is related to how to combine the low-priority components $\boldsymbol{v}_{d2}$ and will be detailed in Section IV-D. Meanwhile, the matrix $\boldsymbol{W}_B$ should be determined so that the joint angles are within the limits determined by $\boldsymbol{q}_{\max}$ and $\boldsymbol{q}_{\min}$. Based on Chan and Dubey's [42] work, we set the i-th component $\boldsymbol{W}_B$ in the following manner:

$$W_{B,i} = \begin{cases} \dfrac{(q_{i_{\max}} - q_{i_{\min}})^2}{4(q_{i_{\max}} - q_i)(q_i - q_{i_{\min}})} \\ \qquad \text{if } q_i \in (q_{i_{\min}}, q_{i_{\max}}) \wedge \\ \qquad (q_i - (q_{i_{\min}} + q_{i_{\max}})/2)\dot{q}_i > 0 \\ 1 \qquad \text{otherwise.} \end{cases} \tag{23}$$

IV. MAIN COMPONENTS OF THE CONTROLLER

A. CART-FLYWHEEL-TABLE MODEL & ITS CONTROL

The COM shifter and the body rotator, which are the main components of the proposed controller, are built upon the model (2), which can be referred to as a cart-flywheel-table model. For simplicity, let us consider a two-dimensional version of the model, which can be illustrated as in Fig. 5 and written as follows:

$$\ddot{p} = u_1 \tag{24a}$$
$$\dot{L} = u_2 \tag{24b}$$
$$r = p - u_1/\omega^2 - u_2/W. \tag{24c}$$

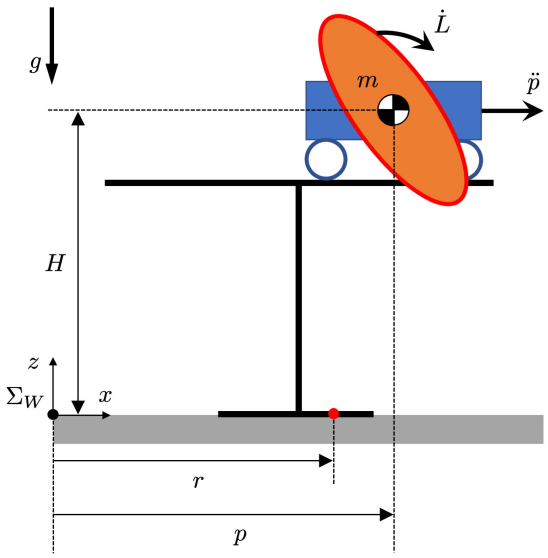

FIGURE 5. The cart-flywheel-table model, in which the cart acceleration $\ddot{p}$ and the flywheel angular momentum rate $\dot{L}$ are treated as inputs and the ZMP r is treated as the output.

Here, p is the cart position, r is ZMP, which resides in the table foot, and L is the angular momentum of the flywheel. The plant parameters are $\omega \triangleq \sqrt{g/H}$ and $W \triangleq mg$ where H is the height of the table and m is the mass of the cart. In this model, the cart acceleration $\ddot{p}$ and the flywheel angular momentum rate $\dot{L}$ are treated as the inputs u_1 and u_2, respectively, and the ZMP r is treated as the output. We also assume that the cart position p and velocity $\dot{p}$ are available to controllers. When the input u_2 is set to zero, (24) reduces to the conventional cart-table model [29]. A similar notion, a cart-table with flywheel model, has been mentioned in [43], where the term u_2 is treated as a perturbation that shrinks the support polygon, in which ZMP should reside. In contrast, here we treat both u_1 and u_2 as control inputs.

To make p and r of the plant (24) converge to a reference ZMP r^{ref}, we consider the following two controllers:

$$u_1 = \text{sat}(a_{\text{lim}}, k_p(r^{\text{ref}} - p) - k_d\dot{p}) \tag{25a}$$
$$u_2 = W(r^{\text{ref}} - \hat{r}) \tag{25b}$$

where

$$\hat{r} \triangleq p - u_1/\omega^2. \tag{26}$$

Here, k_p and k_d are positive controller gains and a_{lim} is the acceleration limit determined by the actuator capacity. The controllers (25a) and (25b) are the basic forms of the COM shifter and the body rotator, respectively, of which the complete forms are presented in the subsequent Sections IV-C and IV-D. We refer to the value $\hat{r}$ as a CT-ZMP because it can be seen as a ZMP value estimated only by the cart-table model, which is (24) with $u_2 \equiv 0$.

It must be noted that the controller (25b), i.e., the body rotator, cannot be always active because it results in the unbounded drift of the angular momentum L, and also in the

unbounded rotation of the robot's body. Therefore, one can see that only the controller (25a) can be always active and that the controller (25b) should be activated only when the error $|r^{\mathrm{ref}} - \hat{r}|$ is large.

With the controllers (25) applied to the plant (24), as long as u_1 is not saturated, the following relations are satisfied:

$$\mathcal{L}[p] = \frac{k_p}{k_p + k_d s + s^2}\mathcal{L}[r^{\mathrm{ref}}] \tag{27}$$

$$\mathcal{L}[\hat{r}] = \frac{k_p(1 - s^2/\omega^2)}{k_p + k_d s + s^2}\mathcal{L}[r^{\mathrm{ref}}] \tag{28}$$

$$\mathcal{L}[r] = \mathcal{L}[\hat{r}] - \mathcal{L}[u_2]/W = \mathcal{L}[r^{\mathrm{ref}}]. \tag{29}$$

The relation (29) shows that the ideal situation $r = r^{\mathrm{ref}}$ is realized with both controllers (25a) and (25b) activated, but as mentioned above, (25b) cannot be always used. It should be noted that, even only with (25a), i.e., with $u_2 \equiv 0$, the relations (27) and (28) are satisfied and also $r = \hat{r}$ is satisfied. Therefore, one needs to tune the controller (25a) to achieve an appropriate response of p and $\hat{r}$ to r^{ref}. A careful observation on the transfer function in (28) reveals that canceling the slower pole by the stable zero $-\omega$ results in a faster, monotonic convergence of $\hat{r}$ to r^{ref}. It can be realized by the setting

$$k_d = k_p/\omega + \omega \ \wedge \ k_p > \omega^2 \tag{30}$$

with which (28) reduces to

$$\mathcal{L}[\hat{r}] = \frac{k_p(1 - s/\omega)}{k_p + \omega s}\mathcal{L}[r^{\mathrm{ref}}]. \tag{31}$$

The COM shifter detailed in the next section is based on this basic idea.

Fig. 6 shows numerical examples of the cart-table model (24) with $u_2 = 0$ combined with the controller (25a) with different gain settings. It can be seen that the controller with $k_d < k_p/\omega + \omega$ leads to faster convergence but overshoot in ZMP. On the contrary, the controller with $k_d > k_p/\omega + \omega$ results in monotonic but slower convergence of the ZMP. The setting (30) realizes a fast and non-overshooting convergence.

Note that the controller (25a) can be seen as a point-to-point controller, as opposed to a trajectory-tracking controller, in the sense that it aims to make both the ZMP r and COM p quickly converge to the reference ZMP r^{ref} without making overshoots, not to always track r^{ref}. Thus, the reference ZMP r^{ref} can discontinuously jump from one point to another, which is always the case in our controller framework. This problem setting is somewhat different from those in [29] and [44], in which the controllers are designed to track continuous trajectories of the reference ZMP.

B. RELATIONS TO PREVIOUS WORK

The controller (25a) with the setting (30), i.e., the basic form of the COM shifter, accepts the reference ZMP input r^{ref} and provides the COM acceleration output. An idea similar to the special gain setting (30) has been utilized in Sugihara's [37] regulator, which accepts a reference COM p^{ref} and provides a ZMP command. It assumes the following LIP-model plant

$$\ddot{p} = \omega^2(p - u_{s1}) \tag{32}$$

where u_{s1} is the given ZMP command. Note that this plant is the inverse system of the plant (24) with $u_2 = 0$. Sugihara's [37] regulator determines the ZMP command u_{s1} to make p converge to the reference COM p^{ref}, keeping u_{s1} within a support polygon $[r_1, r_2]$ while maximizing the region of attraction. It is of the following form:

$$u_{s1} = \max(r_1, \min(r_2, p^{\mathrm{ref}} + k_s(p - p^{\mathrm{ref}}) + b_s\dot{p})) \tag{33}$$

with the feedback gains k_s and b_s satisfying

$$b_s = k_s/\omega \ \wedge \ k_s > 1. \tag{34}$$

As long as u_{s1} is not saturated, the controller (33) applied to the plant (32) results in the following relation:

$$\mathcal{L}[u_{s1}] = \frac{(k_s - 1)(\omega^2 - s^2)}{s^2 + \omega^2 b_s s + \omega^2(k_s - 1)}\mathcal{L}[p^{\mathrm{ref}}], \tag{35}$$

and with the application of the special gain setting (34), it results in a pole-zero cancellation, reducing (35) to the following:

$$\mathcal{L}[u_{s1}] = \frac{(k_s - 1)(\omega - s)}{s + \omega(k_s - 1)}\mathcal{L}[p^{\mathrm{ref}}]. \tag{36}$$

The pole-zero cancellation is not explicitly mentioned in [37], but it contributes to the monotonic behavior of the command ZMP u_{s1}, minimizing the chance of deviation of u_{s1} from the support polygon. Thus, in a sense, the COM shifter (25a) can be said to be a reversed version of Sugihara's [37] regulator. It should be noted that the idea of matching one of the poles to the stable zero $-\omega$ is also found in [45] and [36].

The COM shifter (25a) can be classified as an online ZMP-based walking pattern generator [28, Section 4.4.3], which moves the COM according to the input of the ZMP reference. Kajita et al.'s [29] preview controller is an example of the same class. From our point of view, however, its structure is not very simple, involving a FIFO buffer and a predetermined optimization-based series of gains. There have been many improved methods such as those based on the model predictive control [32], [35] and those with automatic generation of COM reference trajectories [36], [45]. These methods are not straightforward to apply to step-by-step teleoperation, in which future reference values are not available.

The angular momentum rate, which appears in the body rotator (25b), has been taken into account in many previous studies [13], [30], [32], [38], [46], [47]. Many of them involve time-series generators or gait planners, which are not very feasible for step-by-step teleoperation. An idea similar to (25b) has been found in [30], in which the "shortage" of the ZMP calculated from LIP is compensated by a torque around the COM.

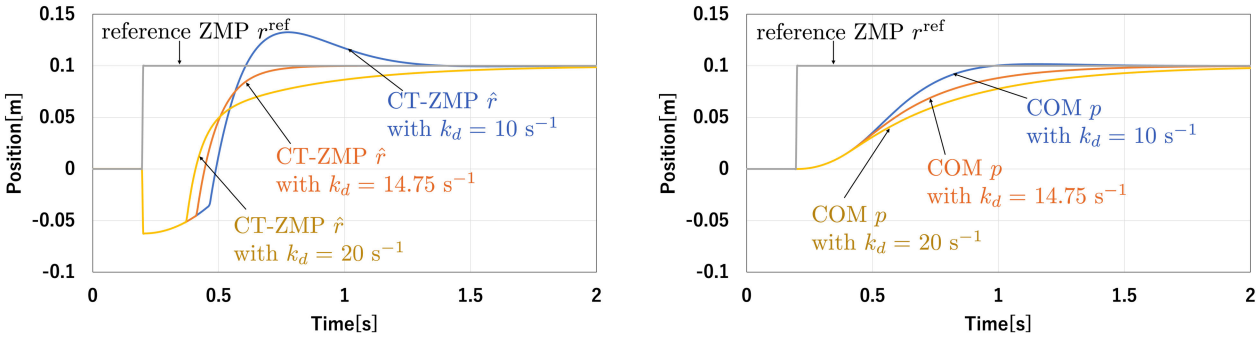

FIGURE 6. Some numerical examples of the controller (25a) applied to the cart-table model (24) with $u_2 \equiv 0$. Trajectories of (a) $\hat{r}$ and (b) p with $a_{\mathrm{lim}} = 0.8$ m/s^2, $\omega = 3.58$ s^{-1}, $k_p = 40$ s^{-2}, and different k_d values. The value $k_d = 14.75$ s^{-1} satisfies (30). The values of a_{lim}, ω and k_p are the same as those in the simulations in Section V.

C. IMPLEMENTATION OF COM SHIFTER

The COM shifter, one of the main components of the proposed controller, is realized by the basic idea of (25a). As shown in Fig. 3, it receives the reference ZMP $\boldsymbol{r}_{xy}^{\mathrm{ref}}$ and generates the desired COM acceleration $\boldsymbol{a}_{Gdxy}$. It is defined as

$$\begin{cases} a_{Gdx} = \mathrm{sat}(a_{\mathrm{lim}}, k_{px}(r_x^{\mathrm{ref}} - p_{Gx}) - k_{dx}\dot{p}_{Gx}) \\ a_{Gdy} = \mathrm{sat}(a_{\mathrm{lim}}, k_{py}(r_y^{\mathrm{ref}} - p_{Gy}) - k_{dy}\dot{p}_{Gy}) \end{cases} \tag{37}$$

with the gain settings

$$k_{d*} = k_{p*}/\omega + \omega \qquad k_{p*} > \omega^2 \tag{38}$$

where $* \in \{x, y\}$ and $\omega \triangleq \sqrt{g/p_{Gz}^{\mathrm{ref}}}$. The desired velocity $\boldsymbol{v}_{Gdxy}$ to be provided to PDIK is obtained by the simple time integration of $\boldsymbol{a}_{Gdxy}$. The controller parameter a_{lim} should be chosen based on the hardware capacity of the actuators and the robot foot size to ensure that ZMP will not exceed the support polygon. In the robot in the simulations in Section V, the parameters were set as $a_{\mathrm{lim}} = 0.8$ m/s^2, $k_{px} = k_{py} = 40$ s^{-2}, $\omega = 3.58$ s^{-1}, and $k_{dx} = k_{dy} = 14.75$ s^{-1} according to (38).

D. IMPLEMENTATION OF BODY ROTATOR

The other main component of the proposed controller, i.e., the body rotator, is built on (25b) presented in Section IV-A. As shown in Fig. 3, the body rotator determines the desired angular momentum rate $\dot{\boldsymbol{L}}_{Gdxy}$, which is integrated into the desired angular momentum $\boldsymbol{L}_{Gdxy}$ that is sent to PDIK. Here, one concern is that it can result in unbounded body rotation due to the time integration. Our solution is to use another signal $\boldsymbol{\omega}_{Bd}$ to keep the torso upright and to prioritize $\boldsymbol{L}_{Gdxy}$ only when the ZMP error is large. This prioritization is realized by changing the weight matrix $\boldsymbol{W}_A$ in PDIK detailed in Section III-D.

Our idea is that the body rotator should be used only when the ZMP error is large in the single-support modes. This idea is realized by determining the desired angular momentum rate $\dot{\boldsymbol{L}}_{Gdxy}$ and the weight matrix $\boldsymbol{W}_A$ as follows: if SL $\vee$ SR:

$$\begin{cases} \dot{L}_{Gdx} = mg(r_y^{\mathrm{ref}} - (p_{Gy} - a_{Gdy}/\omega^2)) \\ \dot{L}_{Gdy} = -mg(r_x^{\mathrm{ref}} - (p_{Gx} - a_{Gdx}/\omega^2)) \end{cases} \tag{39a}$$

$$\begin{cases} W_{A,1} = \max(w_L(r_y^{\mathrm{ref}} - r_y)^2/L_H^2, \varepsilon) \\ W_{A,2} = \max(w_L(r_x^{\mathrm{ref}} - r_x)^2/L_W^2, \varepsilon) \\ W_{A,3-5} = \varepsilon \\ W_{A,6-8} = w_{S1} \end{cases} \tag{39b}$$

if D $\vee$ TL $\vee$ TR:

$$\begin{cases} \dot{L}_{Gdx} = 0 \\ \dot{L}_{Gdy} = 0 \end{cases} \tag{39c}$$

$$\begin{cases} W_{A,1-2} = \varepsilon \\ W_{A,3-5} = w_B \\ W_{A,6-8} = w_{S2}. \end{cases} \tag{39d}$$

Note that the terms $p_{G*} - a_{Gd*}/\omega^2$ in (39a) are the CT-ZMP from (26), which are calculated from the outputs of the COM shifter. Also note that, from the definition of $\boldsymbol{v}_{d2}$ in (19b), $W_{A,1-2}$ are the weights for the body angular momentum $\boldsymbol{L}_{Gdxy}$, $W_{A,3-5}$ are for the torso angular velocity $\boldsymbol{\omega}_{Bd}$, and $W_{A,6-8}$ are for the swing-foot velocity $\boldsymbol{v}_{Sd}$. The constants L_H and L_W in (39b) are the length and width of the robot foot, respectively. With the robot used in the simulations in Section V, the foot size is $L_H = 0.3$ m and $L_W = 0.24$ m, and we chose the values $\{w_L, w_B, w_{S1}, w_{S2}, \varepsilon\}$ to be $\{400\ \mathrm{kg}^{-2} \cdot \mathrm{m}^{-4}, 100, 30\ \mathrm{rad}^2/\mathrm{m}^2, 100\ \mathrm{rad}^2/\mathrm{m}^2, 0.001\}$. In our preliminary simulations with some robots of different sizes, the above setting achieved fairly acceptable results.

The ideas behind these settings are summarized as follows:

- The output of the body rotator is utilized only in the single-support modes and its weight should be larger when the ZMP error $\|\boldsymbol{r}^{\mathrm{ref}} - \boldsymbol{r}\|$ is large.
- The weight for $\boldsymbol{\omega}_{Bd}$ should be large in the double support phase because, in this phase, the body needs to resume the upright attitude.
- The weights for $\boldsymbol{v}_{Sd}$ are set as $w_{S1} < w_{S2}$ because the position control of the swing foot should be accurate in

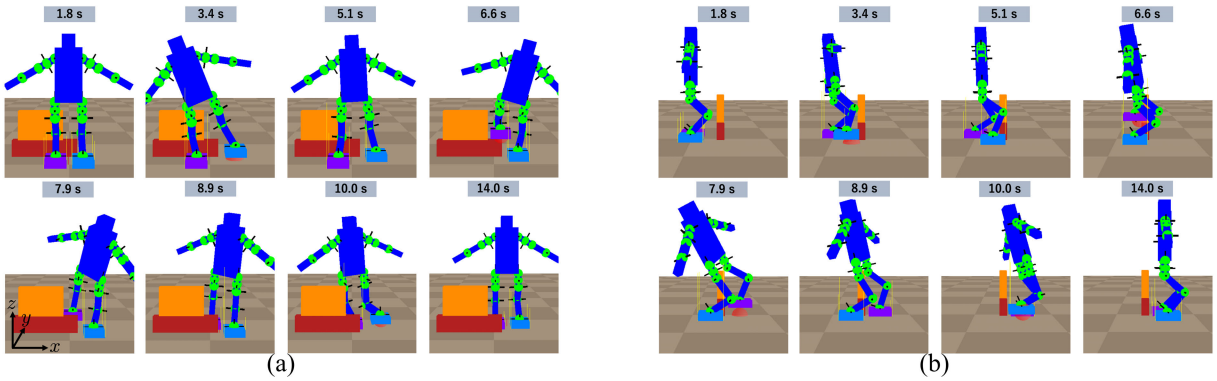

FIGURE 7. Scenario 1: Snapshots of teleoperated bipedal walking with a proper maneuver of the swing foot to avoid an obstacle on the flat terrain. The red sphere indicates the command position p_L^{device} or p_R^{device}. (a) View from the y direction. (b) View from the x direction.

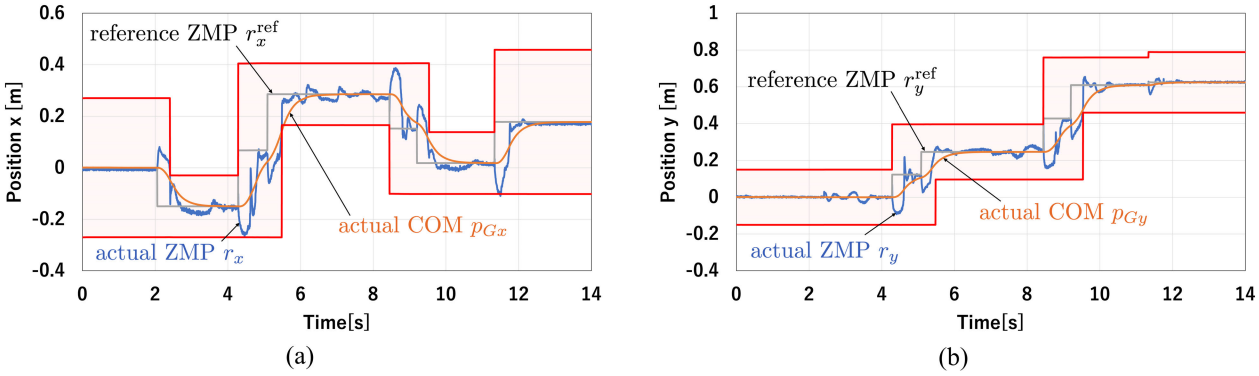

FIGURE 8. Scenario 1: Simulation results of ZMP, COM and the support polygon, which is shown by the red-hatched area. (a) Trajectories in the x direction. (b) Trajectories in the y direction.

the double support phase, to maintain contact with the ground, but can be less accurate in the single support phase to prioritize the automatic balancing.

One imaginable problem may be that the robot body does not resume the upright posture in the single support phase because $W_{A,3-5}$ is ε as in (39b). Setting $W_{A,3-5}$ larger when $\|\boldsymbol{r}^{\text{ref}} - \boldsymbol{r}\|$ is small might be a solution, but it needs a very careful tuning not to hamper the effect of the body rotator. Assuming that robots usually do not keep standing on one leg for a long time, it would not be a big problem. In addition, if necessary, we can allow the user to manually set $W_{A,3-5}$ larger by, e.g., some auxiliary buttons, to compulsorily resume the upright posture. Nevertheless, the body becomes upright once the foot touches down on the ground.

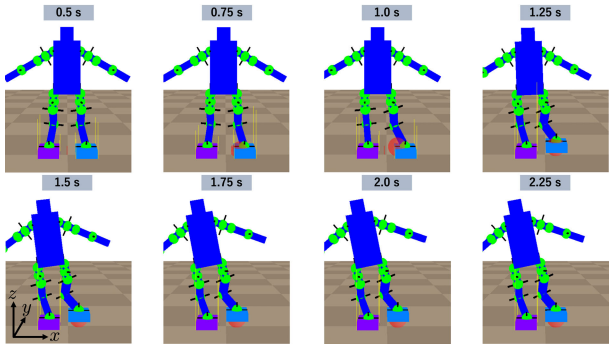

FIGURE 9. Scenario 2: Snapshots of the simulation where a transition takes place from the double support phase to the single support phase through the proposed controller.

V. SIMULATION RESULTS

A. SIMULATION PLATFORM

The proposed controller was validated in the interactive/realtime simulation environment shown in Fig. 1. The simulator was developed by the author's laboratory using Microsoft Visual C++ environment, and was run on a desktop PC with an Intel Core i7-6950X processor (3.00 GHz). In the simulator, the contact forces between the robot and environment were simulated through a penalty-based frictional contact model proposed in [48] and [49]. The timestep size for the physics simulation was set as 0.001 s and the sampling interval of the controller was set as $T = 0.005$ s. We used two Novint Falcons to send position commands $\boldsymbol{p}_L^{\text{device}}$ and $\boldsymbol{p}_R^{\text{device}}$ without force feedback. For the reproducibility of the results, the experimenter moved the falcons by hands, the commands $\boldsymbol{p}_L^{\text{device}}$ and $\boldsymbol{p}_R^{\text{device}}$ were saved in data files, and the saved sequences of $\boldsymbol{p}_L^{\text{device}}$ and $\boldsymbol{p}_R^{\text{device}}$ were replayed in each scenario of the simulation.

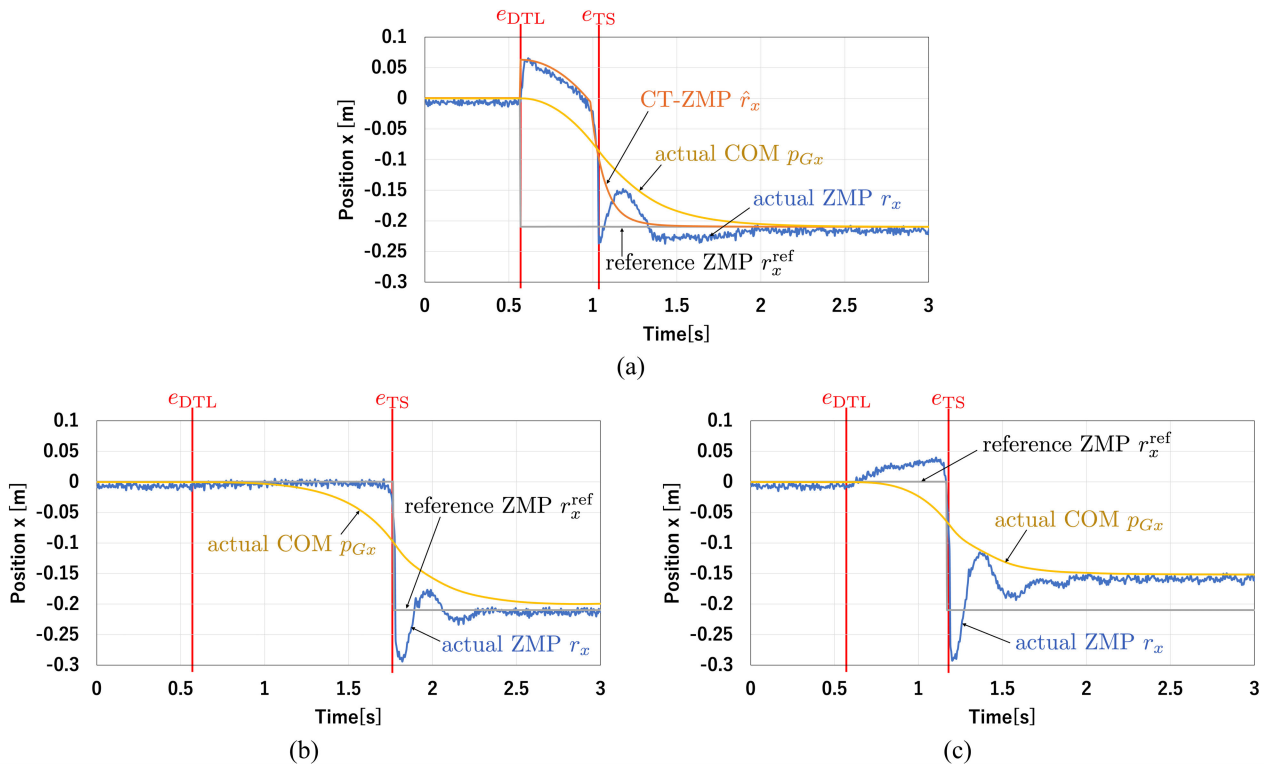

FIGURE 10. Scenario 2: Simulation results of transition from the mode D to the mode SL with (a) the proposed controller and with the preview control with (b) $N = 240$ and (c) $N = 120$. The event e_{DTL} is made happen at $t = 0.57$ s by lifting the right pointing device. The event e_{TS} indicates the lifting of the right foot. The CT-ZMP $\hat{r}_x$ stands for the ZMP value calculated through (26).

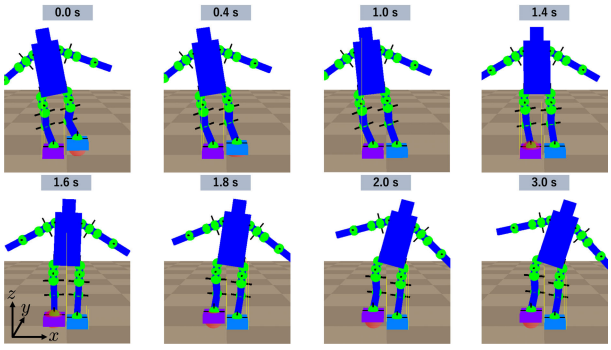

FIGURE 11. Scenario 3: Snapshots of a simulation of mode transitions from SL via D and TR to SR.

The total mass of the teleoperated robot was 65 kg, the height was 1.62 m, and the foot size was 0.3 m×0.24 m. The robot had 20 DOFs in total, including 6 DOFs in each leg and 4 DOFs in each arm. The two arms were controlled to maintain a constant posture. The robot in the simulator was assumed to be equipped with angle sensors attached to the joints, load cells mounted at the four corners of each foot sole to measure the actual ZMP $\boldsymbol{r}$, and a 3-axis gyro sensor to measure the torso attitude. The load cells were simulated with 1000 N of full scale and 0.5% full scale of accuracy, which is realized by adding randomly generated noise to the true value. We set some errors in the robot's dynamics model in the controller; the error in the mass of each link was set within 1% of the true value, and the error in the COM position of each link in each direction was set within 5% of the dimensions of the link in the direction. The errors were randomly generated and kept constant throughout all scenarios of this section.

B. SCENARIO 1: TELEOPERATED BIPEDAL WALKING ACROSS AN OBSTACLE

In Scenario 1, we simulated the step-by-step teleoperation of the bipedal robot with a proper maneuver of the swing foot to avoid an obstacle on the flat terrain. Fig. 7 shows snapshots of the simulation, in which the robot makes two steps (right and then left) across an obstacle (composed of two blocks, brownish red and orange), and eventually re-aligns the feet. The red sphere in each snapshot indicates the command position $\boldsymbol{p}_L^{\mathrm{device}}$ or $\boldsymbol{p}_R^{\mathrm{device}}$, which are sent from the pointing devices operated by the experimenter. As can be seen in Fig. 7, the right foot was moved right-forward and the left foot was lifted high to avoid the obstacles. These motions were performed by the experimenter, who carefully manipulated them not to make the swing foot collide with the obstacle. It illustrates the benefit of the step-by-step teleoperation scheme and the proposed controller, which allow the user to carefully manipulate the swing foot, albeit it demands a certain level of carefulness and skill from the user.

Another point that should be noted in Fig. 7 is that, during the motion, the torso posture significantly varies to extend the swing foot. This is the effect of the PDIK to extend the range

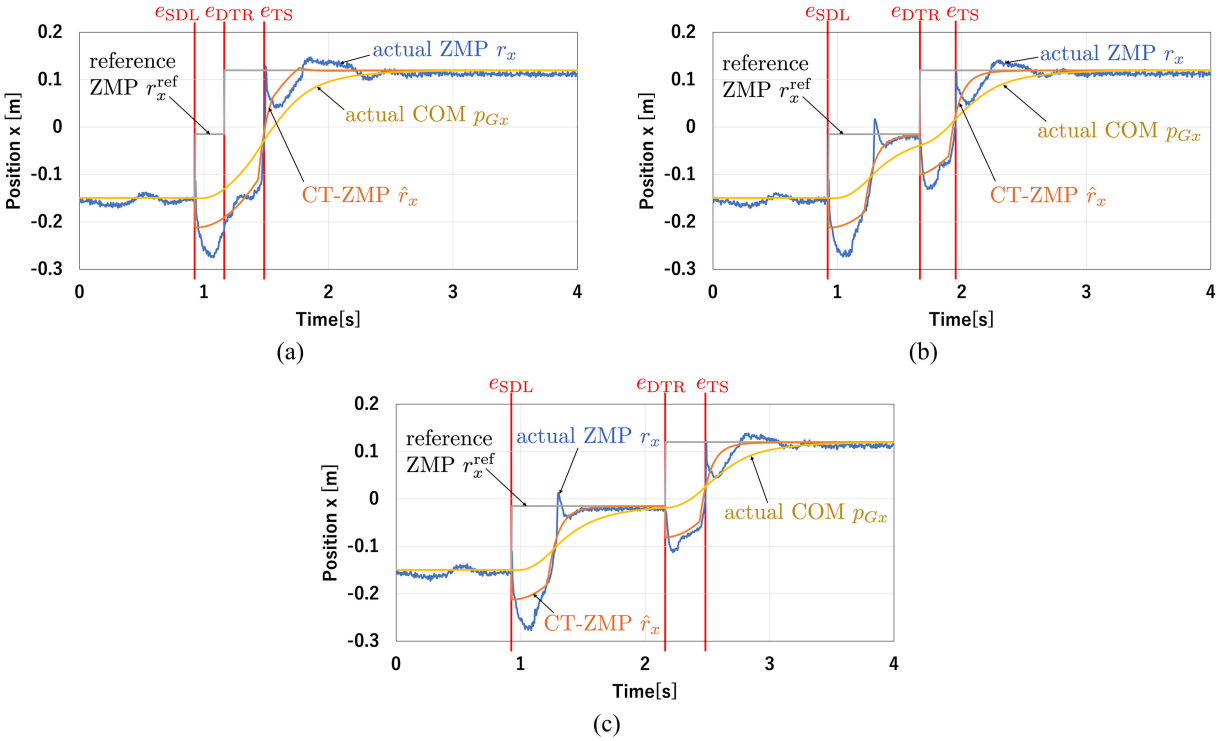

FIGURE 12. Scenario 3: Simulation results of transitions from SL via D and TR to SR with different timing of the event e_{DTR}, which is the lifting of the left pointing device, at (a) $t \approx 1.2$ s, (b) 1.7 s, and (c) 2.2 s. The event e_{SDL} is the grounding of the right foot, which is initiated by moving the right pointing device downward. The event e_{TS} indicates the lifting of the left foot.

of motion of the swing foot within keeping the joint angles within the limits.

Fig. 8 shows the results of ZMP and COM. The red-hatched areas indicate the support polygon, which is determined by the geometry of the feet in contact with the ground. It can be seen that, when the reference ZMP $[r_x^{\text{ref}}, r_y^{\text{ref}}]^T$ changes, the actual ZMP $[r_x, r_y]^T$ first moves in the opposite direction and later it converges to the reference ZMP $[r_x^{\text{ref}}, r_y^{\text{ref}}]^T$. This feature is due to the COM shifter (37), which contributes to the quick shifting of COM. Its effect will be investigated in more detail in Scenario 2. The results also show that the tracking error of ZMP is limited in a small range in the single support phase, which can be attributed to the body rotator. Its effect will be discussed in more detail in Scenario 4.

C. SCENARIO 2: LIFTING ONE FOOT

In Scenario 2, we compared the proposed controller to Kajita et al.'s preview control [29] [28, Section 4.4] in the transitions from the mode D to the mode SL. This comparison aims to clarify that the COM shifter realizes fast ZMP shifting without using FIFO buffers. The event e_{DTL} was made happen at $t = 0.57$ s by lifting the right pointing device, and the reference ZMP r_x^{ref} was changed from 0 m to -0.21 m, which is the location of the left foot. As for the preview control, the weights were set as $Q = 1.0$ and $R = 1.0 \times 10^{-6}$ and the length N of the FIFO was set as 240 and 120 for two simulations (see [28] for definitions). Fig. 9 shows snapshots of the simulation with the proposed controller.

Simulation results are shown in Fig. 10. Fig. 10(a) shows that the proposed controller only took around 0.5 s to lift the foot without causing a stead-state error. In contrast, the preview control with $N = 240$ took $NT = 1.2$ s to lift the right foot, which is too slow for teleoperation, as shown in Fig. 10(b). It can be seen from Fig. 10(c) that the preview control with $N = 120$ took only $NT = 0.6$ s, which is faster than the case of $N = 240$, but leading to a significant steady-state error. As seen from these results, the preview control with a smaller FIFO length N results in a shorter response time but a larger steady-state error. Although there would be some ways to improve it, e.g., [50], the rather complicated structure of the preview control, involving a FIFO buffer, would count as a drawback.

D. SCENARIO 3: SWITCHING OF THE SUPPORT FOOT

In Scenario 3, transitions from the mode SL to the mode SR via the modes D and TR were tested. The transition from the mode SL to the mode D were made by moving the right pointing device downward to ground the right foot, which created the event e_{SDL}. The transition from the mode D to the mode SR via the mode TR were made by moving the left pointing device upward, which created the event e_{DTR}. The reference ZMP r_x^{ref} was changed according to the mode transitions as defined in (15a). Fig. 11 shows snapshots of the simulation.

Simulation results are shown in Fig. 12. In Fig. 12(a), e_{DTR} was given shortly after the event e_{SDL} and thus the COM maintained the maximum acceleration throughout the mode D. In Fig. 12(b), e_{DTR} was given after the actual ZMP

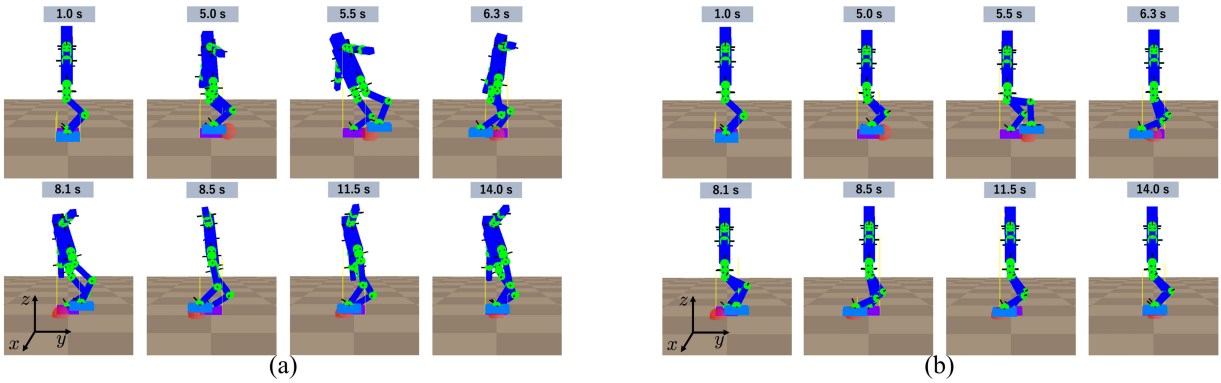

FIGURE 13. Scenario 4: Fast swing of the leg in the single support phase with the body rotator (a) enabled and (b) disabled.

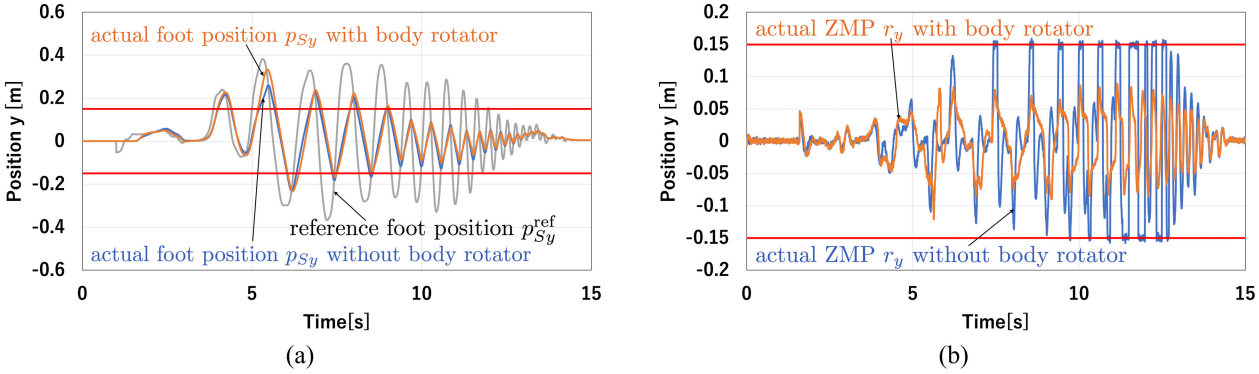

FIGURE 14. Scenario 4: Simulation results of the proposed controller with the body rotator enabled and disabled; (a) The reference position p_{Sy}^{ref} and the actual position p_{Sy} of the foot in the y direction. (b) The actual ZMP r_y in the y direction. The red lines indicate the boundaries of the support polygon in the y direction.

reached the midpoint. In Fig. 12(c), e_{DTR} was given after both ZMP and COM were settled at the midpoint. In all cases with different timings of the trigger event, switching of the support foot was appropriately realized in a responsive manner by the proposed controller.

E. SCENARIO 4: FAST SWING OF A LEG IN SINGLE SUPPORT PHASE

In Scenario 4, simulations of fast motion of the swing foot were performed to compare the cases with the body rotator enabled and disabled. This comparison is for illustrating the effect of the body rotator, which compensates for the oscillation of ZMP caused by swing-foot motions. When the body rotator was disabled, $\dot{\boldsymbol{L}}_{Gdxy}$ was set to be zero and $\boldsymbol{W}_A$ was always set as (39b). Fig. 13 shows snapshots of the simulation with the body rotator enabled and disabled. Fig. 13(a) shows that, with the body rotator, the robot significantly changed its posture as an effect of the body rotator. On the contrary, Fig. 13(b) shows that the torso was kept vertical to the ground when the body rotator is disabled.

Fig. 14 shows the results. Fig. 14(a) shows the reference foot position p_{Sy}^{ref}, which is the common input to both cases, and the resultant foot trajectories p_{Sy} with or without the body rotator. It shows that the foot motions were almost the same between the two cases. Fig. 14(b) shows the ZMP r_y in the two cases. It shows that the fluctuation of the ZMP r_y was made much smaller with the body rotator under almost the same foot motions. These results show that the body rotator is effective to suppress the ZMP error under the disturbance caused by the swing foot motion.

VI. CONCLUSION

The contribution of the paper can be summarized as follows:

- This paper has presented a controller for a step-by-step teleoperation scheme for bipedal robots, in which the user manipulates the foot positions of the robot at every step of walking.
- The controller comprises two main components, a COM shifter and a body rotator, which are combined through prioritized differential inverse kinematics (PDIK) with devised weight matrices.
- The COM shifter is a simple controller to produce a COM motion according to a reference ZMP, and is free from FIFO buffers, time-series generation, or online optimization.
- The body rotator is also a simple controller, which complements the COM shifter to produce an appropriate angular momentum rate to suppress the effect of swing-foot motion induced by rough commands from the user.

The proposed controller has been validated with a realtime simulation environment. The results have shown the advantage of the teleoperation scheme, which allows the user to intuitively realize walking across various obstacles through precise manipulation of the swing foot. It can also be seen from the results that the proposed controller realizes responsive lifting and landing of the feet according to the user commands, and also maintains the balance even under disturbances caused by a fast motion of the swing foot.

Future studies should of course address real hardware experiments to validate the proposed controller, in which the parameter sets used in this simulation study would be useful as an initial setting for fine parameter tuning for the hardware. The comparison between the presented simulation results and hardware experimental results would also give insights into hardware imperfections that could cause potential implementation problems. The controller should also be extended to cope with external forces, more specifically, to be capable of pushing recovery motion [31], [34], [51] and automatic stepping motion [52], [53]. In addition, to deal with uneven terrains, the reference COM height and the reference attitude of the swing foot may need to be varied in adaptive ways. A better set of parameter tuning guidelines should also be sought.

APPENDIX.

The notation $(\ln \boldsymbol{R})^\vee$, which is the combination of the matrix logarithm and the 'vee' operation, represents the conversion from a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ to its corresponding rotation vector (angle-axis representation), which have been used in, e.g., [28, Section 2.2.7]. It is written as follows:

$$(\ln \boldsymbol{R})^\vee = \begin{cases} [0\ 0\ 0]^T & \text{if } \boldsymbol{R} = \boldsymbol{I} \\ \pi[1\ 0\ 0]^T & \text{if } \boldsymbol{R} = \mathrm{diag}(1,-1,-1) \\ \pi[0\ 1\ 0]^T & \text{if } \boldsymbol{R} = \mathrm{diag}(-1,1,-1) \\ \pi[0\ 0\ 1]^T & \text{if } \boldsymbol{R} = \mathrm{diag}(-1,-1,1) \\ \mathrm{atan2}(\|\boldsymbol{l}\|, \mathrm{tr}(\boldsymbol{R})-1)\boldsymbol{l}/\|\boldsymbol{l}\| & \text{otherwise} \end{cases} \tag{40}$$

where $\boldsymbol{l} \triangleq (\boldsymbol{R} - \boldsymbol{R}^T)^\vee$ and $\vee$ is the 'vee' operator, which is defined by $[\boldsymbol{a}\times]^\vee = \boldsymbol{a}$ for all $\boldsymbol{a} \in \mathbb{R}^3$.

ACKNOWLEDGMENT

The authors' C++ source code for the simulator is adapted from Choreonoid [54], an open-source simulator developed by Dr. Shin'ichiro Nakaoka, National Institute of Advanced Industrial Science and Technology (AIST), Japan.

REFERENCES

[1] N. E. Sian, K. Yokoi, S. Kajita, F. Kanehiro, and K. Tanie, "Whole body teleoperation of a humanoid robot development of a simple master device using joysticks," *J. Robot. Soc. Jpn.*, vol. 22, no. 4, pp. 519–527, 2004.

[2] J. Chestnutt, P. Michel, K. Nishiwaki, J. Kuffner, and S. Kagami, "An intelligent joystick for biped control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2006, pp. 860–865.

[3] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo, and S. Ivaldi, "A multimode teleoperation framework for humanoid loco-manipulation: An application for the iCub robot," *IEEE Robot. Autom. Mag.*, vol. 26, no. 4, pp. 73–82, Dec. 2019.

[4] S. Nakaoka, M. Morisawa, K. Kaneko, S. Kajita, and F. Kanehiro, "Development of an indirect-type teleoperation interface for biped humanoid robots," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, Dec. 2014, pp. 590–596.

[5] S. Kohlbrecher, A. Romay, A. Stumpf, A. Gupta, O. Von Stryk, F. Bacim, D. A. Bowman, A. Goins, R. Balasubramanian, and D. C. Conner, "Human-robot teaming for rescue missions: Team ViGIR's approach to the 2013 DARPA robotics challenge trials," *J. Field Robot.*, vol. 32, no. 3, pp. 352–377, May 2015.

[6] R. Cisneros, S. Nakaoka, M. Morisawa, K. Kaneko, S. Kajita, T. Sakaguchi, and F. Kanehiro, "Effective teleoperated manipulation for humanoid robots in partially unknown real environments: Team AIST-NEDO's approach for performing the plug task during the DRC finals," *Adv. Robot.*, vol. 30, no. 24, pp. 1544–1558, Dec. 2016.

[7] S. J. Jorgensen, M. W. Lanighan, S. S. Bertrand, A. Watson, J. S. Altemus, R. S. Askew, L. Bridgwater, B. Domingue, C. Kendrick, J. Lee, and M. Paterson, "Deploying the NASA Valkyrie humanoid for IED response: An initial approach and evaluation summary," in *Proc. IEEE-RAS 19th Int. Conf. Humanoid Robots (Humanoids)*, Oct. 2019, pp. 379–386.

[8] S. J. Jorgensen, M. Wonsick, M. Paterson, A. Watson, I. Chase, and J. S. Mehling, "Cockpit interface for locomotion and manipulation control of the NASA Valkyrie humanoid in virtual reality (VR)," NASA, Washington, DC, USA, Tech. Rep., MSC-27278-1, 2022.

[9] A. Otaran and I. Farkhatdinov, "Walking-in-place foot interface for locomotion control and telepresence of humanoid robots," in *Proc. IEEE-RAS 20th Int. Conf. Humanoid Robots (Humanoids)*, Jul. 2021, pp. 453–458.

[10] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Adv. Robot.*, vol. 24, nos. 5–6, pp. 719–737, 2010.

[11] T. Sugihara and Y. Nakamura, "Boundary condition relaxation method for stepwise pedipulation planning of biped robots," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 658–669, Jun. 2009.

[12] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1^{st} report: Walking gait pattern generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1084–1091.

[13] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method for real-time gait planning for humanoid robots," *Int. J. Humanoid Robot.*, vol. 3, no. 1, pp. 1–19, Mar. 2006.

[14] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time ZMP gait generation and feedback stabilization," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2015, pp. 936–940.

[15] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2014, pp. 279–286.

[16] I. Almetwally and M. Mallem, "Real-time tele-operation and tele-walking of humanoid robot nao using Kinect depth camera," in *Proc. 10th IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, Apr. 2013, pp. 463–466.

[17] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "High speed whole body dynamic motion experiment with real time master-slave humanoid robot system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5835–5841.

[18] D. K. Prasanga, K. Tanida, K. Ohnishi, and T. Murakami, "Simultaneous bipedal locomotion based on haptics for teleoperation," *Adv. Robot.*, vol. 33, nos. 15–16, pp. 824–839, Aug. 2019.

[19] Y. Ishiguro, T. Makabe, Y. Nagamatsu, Y. Kojio, K. Kojima, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit TABLIS," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6419–6426, Oct. 2020.

[20] J. Oh, I. Lee, H. Jeong, and J.-H. Oh, "Real-time humanoid whole-body remote control framework for imitating human motion based on kinematic mapping and motion constraints," *Adv. Robot.*, vol. 33, no. 6, pp. 293–305, Mar. 2019.

[21] S. Wang and J. Ramos, "Dynamic locomotion teleoperation of a reduced model of a wheeled humanoid robot using a whole-body human-machine interface," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1872–1879, Apr. 2022.

[22] J. Ramos and S. Kim, "Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation," *Sci. Robot.*, vol. 4, no. 35, Oct. 2019, Art. no. eaav4282.
[23] J. Oh, O. Sim, B. Cho, K. Lee, and J.-H. Oh, "Online delayed reference generation for a humanoid imitating human walking motion," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 1, pp. 102–112, Feb. 2021.
[24] G. Colin, Y. Sim, and J. Ramos, "Bipedal robot walking control using human whole-body dynamic telelocomotion," 2022, *arXiv:2209.06964*.
[25] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," 2023, *arXiv:2301.04317*.
[26] T. Ando, T. Watari, and R. Kikuuwe, "Master-slave bipedal walking and semi-automatic standing up of humanoid robots," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2020, pp. 360–365.
[27] T. Ando, T. Watari, and R. Kikuuwe, "Reference ZMP generation for teleoperated bipedal robots walking on non-flat terrains," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2021, pp. 780–794.
[28] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics* (Springer Tracts in Advanced Robotics), vol. 101. Berlin, Germany: Springer, 2014.
[29] 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/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2003, pp. 1620–1627.
[30] Y. Kojio, Y. Ishiguro, K.-N.-K. Nguyen, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Unified balance control for biped robots including modification of footsteps with angular momentum and falling detection based on capturability," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 497–504.
[31] R. Schuller, G. Mesesan, J. Englsberger, J. Lee, and C. Ott, "Online centroidal angular momentum reference generation and motion optimization for humanoid push recovery," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5689–5697, Jul. 2021.
[32] J. Ding, C. Zhou, S. Xin, X. Xiao, and N. G. Tsagarakis, "Nonlinear model predictive control for robust bipedal locomotion: Exploring angular momentum and CoM height changes," *Adv. Robot.*, vol. 35, no. 18, pp. 1079–1097, Sep. 2021.
[33] K. Yamamoto, T. Kamioka, and T. Sugihara, "Survey on model-based biped motion control for humanoid robots," *Adv. Robot.*, vol. 34, nos. 21–22, pp. 1353–1369, Nov. 2020.
[34] 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 (Humanoids)*, Mar. 2006, pp. 200–207.
[35] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *Proc. 6th IEEE-RAS Int. Conf. Humanoid Robots*, Dec. 2006, pp. 137–142.
[36] S. Hong, Y. Oh, D. Kim, and B.-J. You, "Real-time walking pattern generation method for humanoid robots by combining feedback and feedforward controller," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 355–364, Jan. 2014.
[37] T. Sugihara, "Standing stabilizability and stepping maneuver in planar bipedalism based on the best COM-ZMP regulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 1966–1971.
[38] K. Guan, K. Yamamoto, and Y. Nakamura, "Virtual-mass-ellipsoid inverted pendulum model and its applications to 3D bipedal locomotion on uneven terrains," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1401–1406.
[39] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2003, pp. 1644–1650.
[40] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, Jun. 1987.
[41] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *J. Dyn. Syst., Meas., Control*, vol. 108, no. 3, pp. 163–171, Sep. 1986.
[42] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, Apr. 1995.
[43] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping, and whole-body control," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1635–1648, Dec. 2020.
[44] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online decision of foot placement using singular LQ preview regulation," in *Proc. 11th IEEE-RAS Int. Conf. Humanoid Robots*, Oct. 2011, pp. 13–18.
[45] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4489–4496.
[46] S.-H. Lee and A. Goswami, "Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 4667–4672.
[47] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Auto. Robots*, vol. 33, no. 4, pp. 399–414, Nov. 2012.
[48] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto, "Admittance and impedance representations of friction based on implicit Euler integration," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1176–1188, Dec. 2006.
[49] R. Kikuuwe and H. Fujimoto, "Incorporating geometric algorithms in impedance- and admittance-type haptic rendering," in *Proc. 2nd Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst. (WHC)*, Mar. 2007, pp. 249–254.
[50] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, and H. Hirukawa, "Biped walking pattern generator allowing auxiliary ZMP control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 2993–2999.
[51] D. N. Nenchev and R. Iizuka, "Emergent humanoid robot motion synergies derived from the momentum equilibrium principle and the distribution of momentum," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 536–555, Feb. 2022.
[52] M. Morisawa, F. Kanehiro, K. Kaneko, N. Mansard, J. Sola, E. Yoshida, K. Yokoi, and J. Laumond, "Combining suppression of the disturbance and reactive stepping for recovering balance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 3150–3156.
[53] T. Sugihara, "Reflexive step-out control superposed on standing stabilization of biped robots," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2012, pp. 741–746.
[54] S. Nakaoka, "Choreonoid: Extensible virtual robot environment built on an integrated GUI framework," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2012, pp. 79–85.

YACHEN ZHANG received the B.E. degree in mechanical engineering from North China University of Science and Technology, Hebei, China, in 2017, and the M.E. degree in mechanical engineering from Okayama University, Okayama, Japan, in 2021. He is currently pursuing the Ph.D. degree with Hiroshima University, Hiroshima, Japan. His research interests include the whole-body motion planning of the humanoid robot, teleoperation, and control engineering.

RYO KIKUUWE (Member, IEEE) received the B.S., M.S., and Ph.D. (Eng.) degrees in mechanical engineering from Kyoto University, Kyoto, Japan, in 1998, 2000, and 2003, respectively.

From 2003 to 2007, he was an Endowed-Chair Research Associate with Nagoya Institute of Technology, Nagoya, Japan. From 2007 to 2017, he was an Associate Professor with the Department of Mechanical Engineering, Kyushu University, Fukuoka, Japan. From 2014 to 2015, he was a Visiting Researcher with Inria, Grenoble, Rhône-Alpes, Saint-Ismier, France. He is currently a Full Professor with the Graduate School of Advanced Science and Engineering, Hiroshima University, Higashihiroshima, Japan. His research interests include the force control of robot manipulators, real-time simulation for physics-based animation, and the engineering applications of differential inclusions.