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

Yachen Zhang and Ryo Kikuuwe

***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 hand-held 3-degree-of-freedom haptic devices. This teleoperation scheme allows users to precisely manipulate the swing foot motions to traverse rough terrains by avoiding obstacles. The scheme requires a controller that quickly responds to the user commands and maintains the balance 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 built upon a cart-flywheel-table model of bipedal robots. The COM shifter is a simple feedback controller to produce a COM motion according to a reference ZMP (zero moment point). The body rotator is a complement for 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.**

Fig. 1. Interactive simulation setup for the step-by-step teleoperation of a humanoid robot

I. INTRODUCTION

THE application of autonomous humanoid 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 humanoid robots, walking is one of the most important and basic tasks. Because a humanoid robot is an intrinsically unstable mechanism with many degrees of freedom (DOFs) and is prone to falling while walking, an appropriate combination of automatic control and manual control is important for teleoperated bipedal walking.

The majority of studies on the teleoperation of humanoid robots are based on graphical user interfaces (GUIs) [1], [2], [3] on personal computers or joysticks [4], [5], [6] combined with automatic footstep planning techniques. Some researchers choose mechanical ways to map the human operator's motion to the robot's motion. Such approaches employ complex mechatronic devices that restrain the operator's body, such as motion capture systems [7], [8], exoskeletons [9], [10], and other complex devices [11], [12].

Even with sophisticated whole-body interface devices, practical teleoperation is not straightforward because the exact position matching of body parts of the operator and the robot may cause the loss of balance of the robot. Matching the center of mass (COM) and zero moment point (ZMP) between the operator and the robot enables dynamic and intuitive teleoperation [12], 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 is important not to cause the collision of the foot and the environment. Furthermore, it would be better for the operator to use his/her hands, instead of feet, to carefully operate the robot's feet. As far as the authors are aware, there have been no such methods for teleoperated bipedal walking that allows precise manipulation of the swing foot.

This paper proposes a controller for intuitively teleoperating a humanoid robot through a pair of inexpensive, hand-held 3-DOF haptic devices, as in Fig. 1. The two haptic devices correspond to the two feet of the robot, and are used only as pointing devices without force feedback. The controller is for what we call a step-by-step teleoperation scheme, in which the user manipulates the swing-foot motion at every step of walking. In the double support phase, lifting one haptic 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 haptic 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. This scheme has been introduced in the previous publications [13], [14] from our research group, in which some preliminary control techniques have been proposed. This

The corresponding author is Yachen Zhang (e-mail: yachenzhang@mdl.hiroshima-u.ac.jp).

Yachen Zhang and Ryo Kikuuwe are with Machinery Dynamics Laboratory, Hiroshima University, Hiroshima, Japan.

scheme would be seen as advantageous over previous schemes such as [7], [8], [9], [10], [11], [12] in terms of the hardware cost and the physical burden to the user, and also in terms of the maneuverability of the swing foot for traversing obstacles. This scheme would be useful as a complement for automatic gait planning techniques, which would be more efficient for flat or structured environments. A potentially useful scenario is that, when the environment is found to be too complex for automatic gait planners, this scheme could be activated to entrust the motion planning to a human operator.

The step-by-step teleoperation scheme requires the robot to follow unpredictable commands from the user as opposed to predetermined motion patterns. The scheme is therefore incompatible with automatic gait planning techniques (e.g., [15], [16], [17], [18], [19]). Instead, the scheme demands a simple feedback controller capable of quickly responding to the user commands and maintaining the balance even under rough commands from the user. Moreover, to deal with unpredictable user commands, the controller should preferably be a simple feedback controller without involving time-series generators or online optimizers.

In step-by-step teleoperation, when a foot is commanded to be lifted, the desired ZMP should be set in the other foot, and in the single support phase, the desired ZMP should be in the support foot. This means that the step-by-step teleoperation scheme needs a so-called ZMP-based motion generation [20, Section 4.4], with which the user command determines the desired ZMP, and the COM of the robot should be moved accordingly. One example of such controllers is Kajita et al.'s [21] preview controller. Its structure is, from our point of view, not very simple, involving a FIFO (first-in first-out) buffer and a predetermined optimization-based series of gains. There have been many improved methods such as those based on the model predictive control [22], [23] and those with automatic generation of COM reference trajectories [24], [25]. These methods are not straightforward to apply to step-by-step teleoperation, in which future reference values are not available.

Another important feature of step-by-step teleoperation is that the swing leg manipulated by the user can result in significant variation in the angular momentum of the whole body. The angular momentum rate (the time derivative of the angular momentum) also affects the ZMP [18], [23], [26], [27], [28] but its effect is neglected in the linear inverted pendulum (LIP) model [21], on which the majority of bipedal robot controllers, e.g., [16], [17], [19], [25], [29], are built. In addition, there are many techniques considering the angular momentum rate [18], [23], [30], [31] for automatic bipedal walking, but again, they involve time-series generators or gait planners, which are not very feasible for step-by-step teleoperation.

The controller proposed in this paper has the structure inherited from our previous controller [13], [14]. The main improvements are two new components; a COM shifter and a body rotator, both of which are simple feedback controllers free from FIFO buffers, time-series generators, or online optimization. The COM shifter is based on the conventional cart-table model [21] and it can be seen as a reversed version of Sugihara's [29] regulator based on the LIP model. Compared with our previous controller [13], [14], the COM shifter realizes fast ZMP shifting with much simple structure. The body rotator is a complement for the COM shifter to produce an appropriate angular momentum rate to enhance the regulation of ZMP. The body rotator enables the robot to maintain balance even under rough user commands, which is the capacity our previous controller [13], [14] lacks. The proposed controller is validated in our interactive/realtime simulation environment shown in Fig. 1.

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. Conclusions are provided in Section VI.

II. Preliminaries

A. Coordinate frames

We consider a humanoid 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 subscripts G corresponds to the COM of the robot. The subscript S corresponds to the swing foot in single support phase and the right foot in double support phase. In this paper, unless otherwise specified, all vectors of position, velocity, angular velocity and momentum are represented in the world coordinate system Σ_W.

B. The 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 [18], [23], [28]:

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

Here, the subscripts x, y, and z stands 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 assumption is obtained as follows:

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

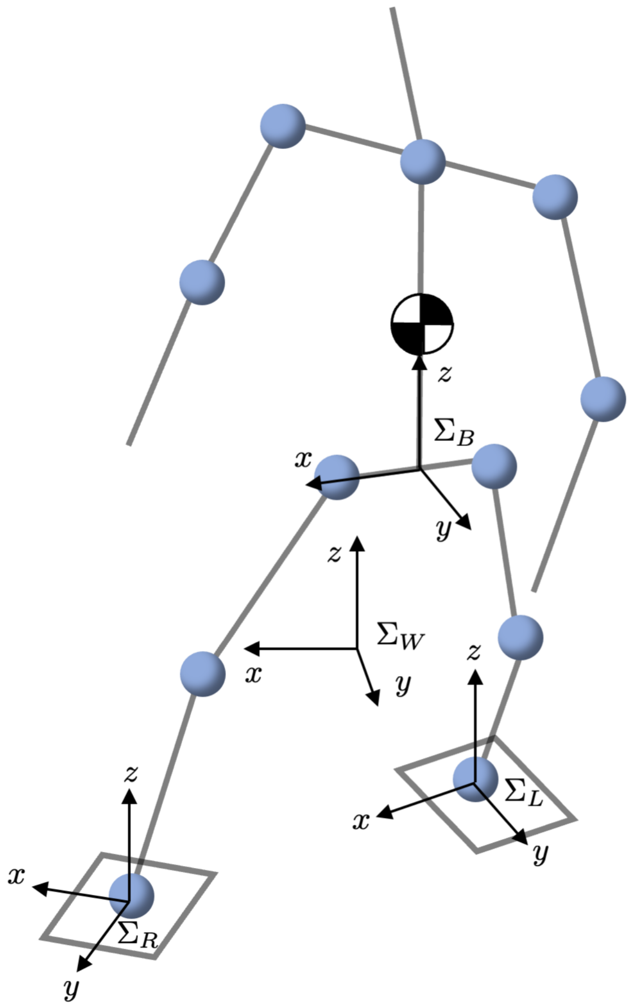

Fig. 2. Coordinate frames associated with a humanoid robot

This formulation corresponds to the linear inverted pendulum plus flywheel model [32] 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 [21]. Some comprehensive discussions on the structure (2), involving $\dot{\boldsymbol{L}}_G$, are found in [28].

The majority of the previous techniques, e.g., [14], [16], [17], [19], [21], [22], [25], [29], 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 [18], [23], [30], [31], many of which depend on predetermined motion commands. This paper builds a simple feedback 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 the 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 [33].

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:

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

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:

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

where the subscript $*$ can be L or R corresponding to the left foot or the right foot, $\boldsymbol{p}_{*B} \in \mathbb{R}^3$ 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:

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

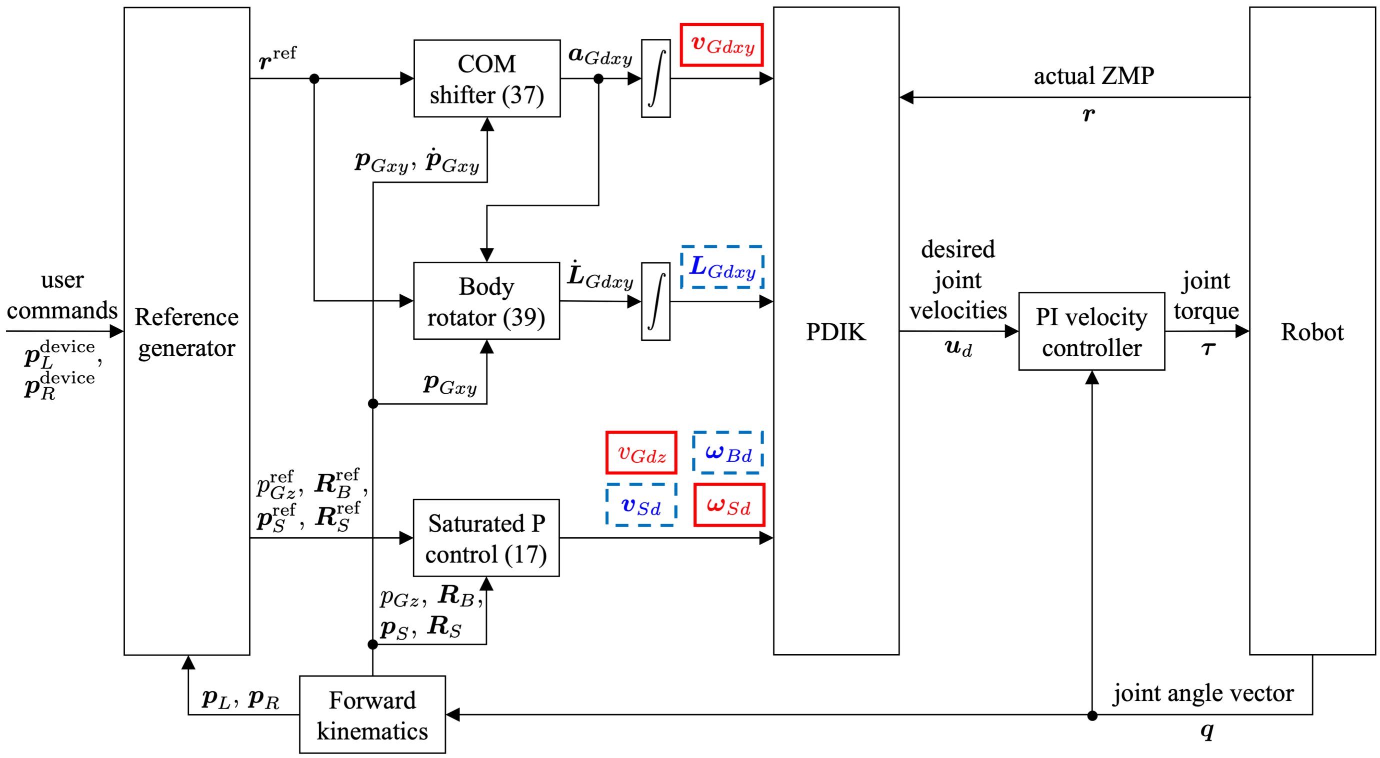

Fig. 3. Block diagram of the proposed controller. The components of the desired velocity vector $v_d \in \mathbb{R}^{14}$ are indicated by red solid boxes and blue dashed boxes, which correspond to high- and low-priority components, respectively

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

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

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

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

where

$$\begin{bmatrix}\hat{M}_G^*\\ H_G^*\end{bmatrix} \triangleq \begin{bmatrix}\hat{M}_B^* + \hat{M}_L & \hat{M}_R & \hat{M}_o\\ H_B^* + H_L & H_R & H_o\end{bmatrix} \quad (13a)$$

$$\begin{aligned}\begin{bmatrix}\hat{M}_B^*\\ H_B^*\end{bmatrix} &\triangleq -\begin{bmatrix} I & -[p_{GB}\times]\\ 0 & \tilde{H}\end{bmatrix}\begin{bmatrix} I & -[p_{LB}\times]\\ 0 & I\end{bmatrix} J_L\\ &= -\begin{bmatrix} I & -[(p_{GB}+p_{LB})\times]\\ 0 & \tilde{H}\end{bmatrix} J_L. \end{aligned} \quad (13b)$$

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 $\hat{M}_G^*$ and H_G^* are Jacobian-like matrices that transform the joint velocities to $\dot{p}_G$ and 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. The structure is mainly inherited from our prior work [13], [14] but is improved in some aspects. The main improvements consist of two new components, a COM shifter and a body rotator, which will be detailed in the next Section IV. This section focuses on the framework to which these new components are incorporated, which has also been improved.

Through this scheme, the robot's feet are manipulated in realtime by the user. More precisely, as shown in Fig. 1, there are two devices, the left-hand and the right-hand haptic devices, which manipulate the left foot and the right foot, respectively. In the double support phase, lifting one haptic device leads to its corresponding foot being lifted. In the single support phase, the swing foot follows the motion of its correspondent haptic device until the sensors detect its contact with the ground.

The robot's joints are assumed to be velocity-controlled with independent servo controllers. Thus, we construct here a controller that sends angular velocity commands to the joints. We only consider the joints of two legs, of which the joint angles are aggregated into the joint angle vector $q \triangleq [q_L^T, q_R^T]^T \in \mathbb{R}^{12}$, and other joints are set with constant angles. The correspondent desired joint velocity vector is denoted by $u_d \in \mathbb{R}^{12}$.

The controller has five different modes as illustrated in Fig. 4 and detailed in Section III-B. The reference generator, appearing in Fig. 3, generates reference values of ZMP and position/attitude information. The reference values are translated into desired velocity information represented by a vector $v_d \in \mathbb{R}^{14}$, as will be detailed in Section III-C. The desired

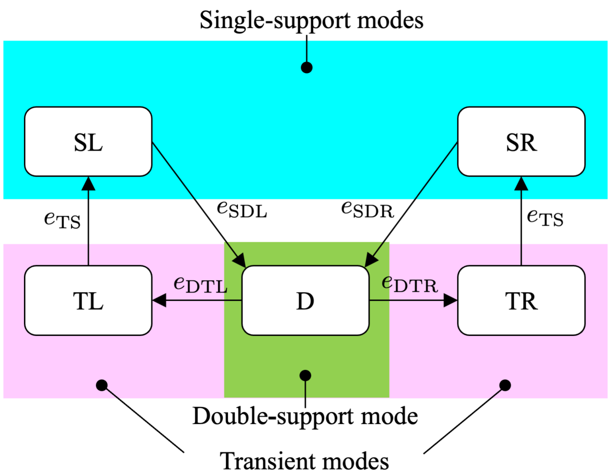

Fig. 4. Mode transition diagram of the proposed controller

velocity vector $\boldsymbol{v}_d$ is converted into joint velocity commands $\boldsymbol{u}_d$ through the PDIK introduced in Section III-D.

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 [14], but there have been some modifications. In Fig. 4, the mode D is the double-support mode, the mode 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.

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

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

Here, the actual ZMP $\boldsymbol{r}$ is measured by load cells at each foot sole, and $\boldsymbol{r}^{\mathrm{ref}} \in \mathbb{R}^3$ 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 haptic device positions $\boldsymbol{p}_L^{\mathrm{device}}$ and $\boldsymbol{p}_R^{\mathrm{device}}$ 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.

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

The reference generator sends the following five quantities:

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

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 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 $\boldsymbol{R}_* \in \mathbb{R}^{3\times 3}$ ($* \in \{S, B\}$) 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:

$$v_{Gdz} = \mathrm{sat}(v_{\lim}, k_v(p_{Gz}^{\mathrm{ref}} - p_{Gz})) \tag{17a}$$
$$\boldsymbol{\omega}_{Bd} = \mathrm{sat}(\omega_{\lim}, k_\omega(\ln \boldsymbol{R}_B^{\mathrm{ref}} \boldsymbol{R}_B^T)^\vee) \tag{17b}$$
$$\boldsymbol{v}_{Sd} = \mathrm{sat}(v_{\lim}, k_v(\boldsymbol{p}_S^{\mathrm{ref}} - \boldsymbol{p}_S)) \tag{17c}$$
$$\boldsymbol{\omega}_{Sd} = \mathrm{sat}(\omega_{\lim}, k_\omega(\ln \boldsymbol{R}_S^{\mathrm{ref}} \boldsymbol{R}_S^T)^\vee) \tag{17d}$$

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

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

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 A.

The velocity limits $v_{\lim}$ and $\omega_{\lim}$ 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 $\{v_{\lim}, \omega_{\lim}, k_v, k_\omega\} = \{1 \text{ m/s}, 0.8 \text{ rad/s}, 10 \text{ s}^{-1}, 10 \text{ s}^{-1}\}$ 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 $\boldsymbol{q} \in \{\boldsymbol{x} \in \mathbb{R}^{12} \mid \boldsymbol{q}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{q}_{\max}\}$ 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 [34].

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 velocity, 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, otherwise the foot will move and ground in an unexpected posture.
- The angular momentum $\boldsymbol{L}_{Gdxy}$ are allowed to be realized with low accuracy due to their relatively low effect on balance in 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 single support phase. Otherwise, the robot will lose balance.

Generating the joint velocity command $\boldsymbol{u}_d$ based on $\boldsymbol{v}_{d1}$ and $\boldsymbol{v}_{d2}$ can be described as the following constrained quadratic optimization problem:

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

where $\boldsymbol{J}_1 \in \mathbb{R}^{6\times 12}$ and $\boldsymbol{J}_2 \in \mathbb{R}^{8\times 12}$ 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 $\boldsymbol{W}_A \in \mathbb{R}^{8\times 8}$ and $\boldsymbol{W}_B \in \mathbb{R}^{12\times 12}$ 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

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

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 (SR-inverse) [35] 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-C. 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 et al.'s [36] work, we set the i-th component $\boldsymbol{W}_B$ in the following manner:

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

IV. Main Components of the Controller

A. Cart-flywheel-table model

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:

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

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

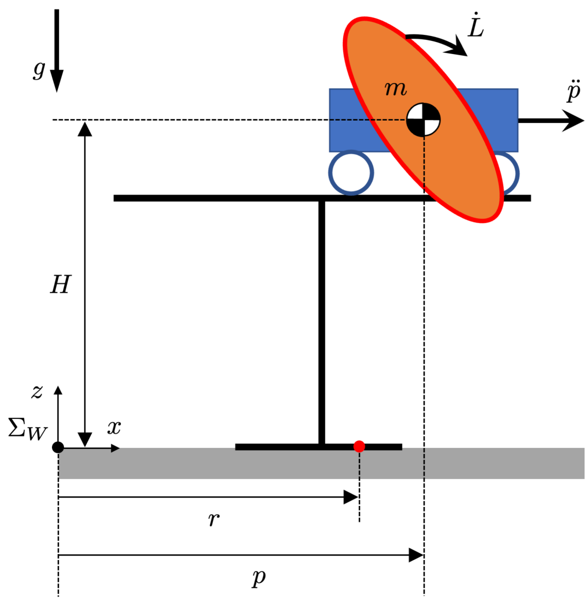

Fig. 5. The cart-flywheel-table model, in which the cart acceleration $\ddot{p}$ and the flywheel angular momentum rate $\dot{L}$ are treated as inputs and the ZMP r is treated as the output

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 input 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 [21]. A similar notion, a cart-table with flywheel model, has been mentioned in [37], in which the term u_2 is treated as a perturbation that shrinks the support polygon in which an estimated 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-B and IV-C. 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 $u_2 \equiv 0$. An idea similar to (25b) has also been found in [31], in which the "shortage" of the ZMP calculated from LIP is compensated by a torque around the COM.

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 $|r^{\text{ref}} - \hat{r}|$ is large.

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

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

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

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

The relation (29) shows that the ideal situation $r = r^{\text{ref}}$ 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

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

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 [21], [38], in which the controllers are designed to track continuous trajectories of the reference ZMP.

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 [29] 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

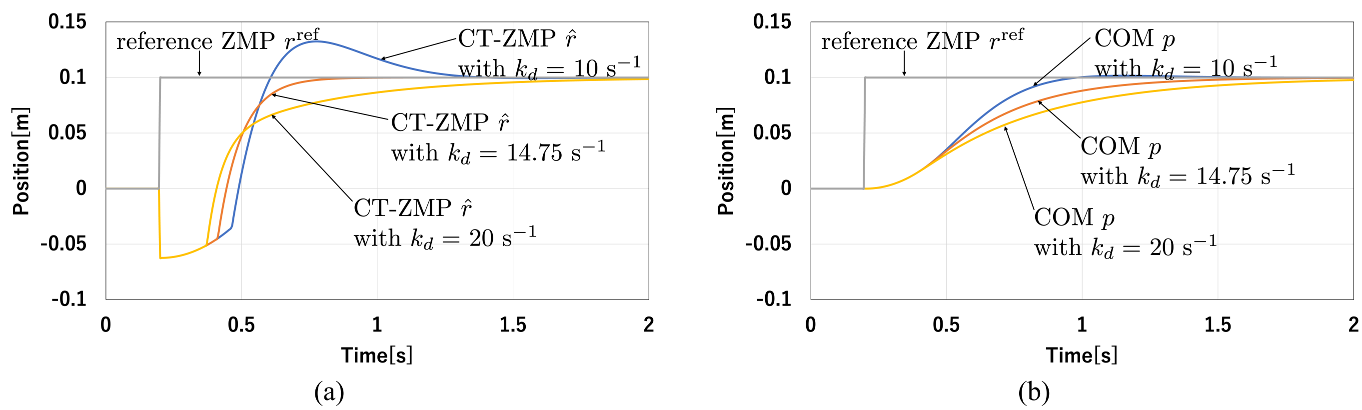

Fig. 6. Some numerical examples of the controller (25a) applied to the cart-table model (24) with $u_2 \equiv 0$. Trajectories of (a) $\hat{r}$ and (b) p with a_{lim} = 0.8 m/s^2, ω = 3.58 s^{-1}, k_p = 40 s^{-2}, and different k_d values. The value k_d = 14.75 s^{-1} satisfies (30). The values of a_{lim},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

[29] 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:

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

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:

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

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

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

The pole-zero cancellation is not explicitly mentioned in [29], 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 [29] regulator. It should be noted that the idea of matching one of the poles to the stable zero $-\omega$ is also found in [24], [25].

B. COM shifter for responsive ZMP shifting

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 $\boldsymbol{r}_{xy}^{\text{ref}}$ and generates the desired COM acceleration $\boldsymbol{a}_{Gdxy}$. It is defined as

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

with the gain settings

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

where $* \in \{x, y\}$ and $\omega \triangleq \sqrt{g/p_{Gz}^{\text{ref}}}$. 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 $a_{\text{lim}} = 0.8$ 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).

C. Body rotator for better ZMP regulating

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:

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

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

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

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

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

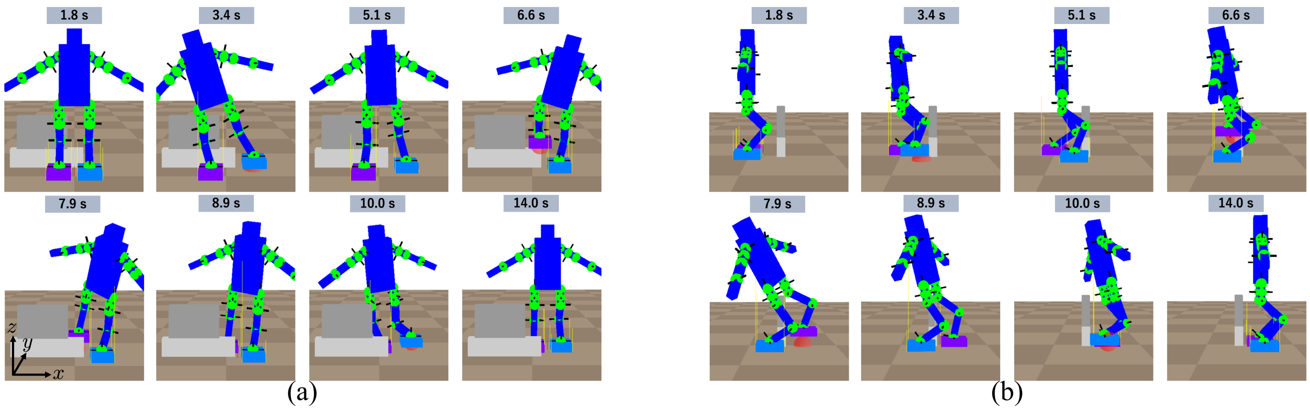

Fig. 7. Scenario 1: Snapshots of teleoperated bipedal walking with a proper maneuver of the swing foot to avoid an obstacle on the flat terrain. The red sphere indicates the command position $\boldsymbol{p}_L^{\rm device}$ or $\boldsymbol{p}_R^{\rm device}$. (a) View from the y direction. (b) View from the x direction

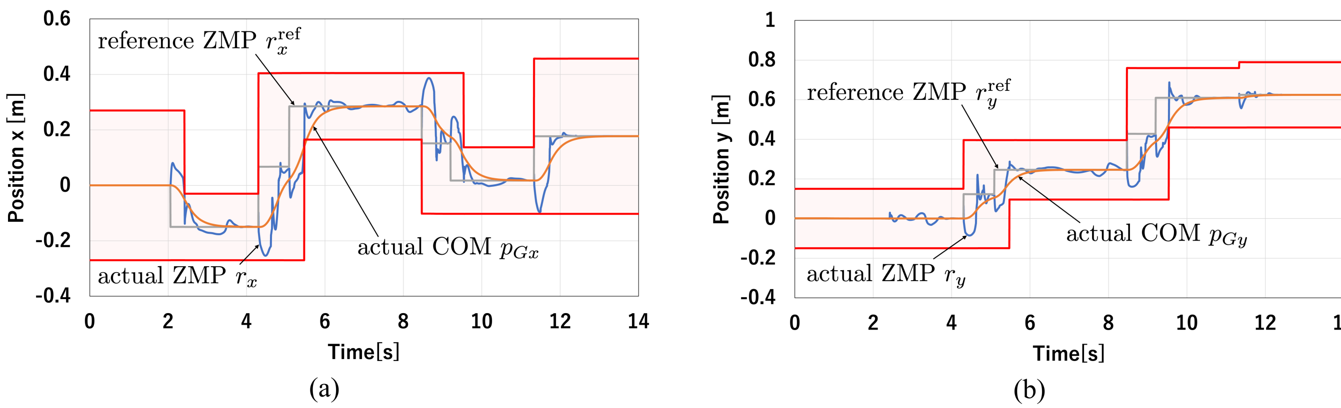

Fig. 8. Scenario 1: Simulation results of ZMP, COM and the support polygon, which is shown by the red-hatched area. (a) Trajectories in the x direction. (b) Trajectories in the y direction

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 with 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 $\|\boldsymbol{r}^{\rm ref} - \boldsymbol{r}\|$ 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 the double support phase, to maintain contact with the ground, but can be less accurate in the single support phase to prioritize the balance.

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 $\|\boldsymbol{r}^{\rm ref} - \boldsymbol{r}\|$ 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.

V. Simulation Results

A. Simulation platform

The proposed controller was validated in the interactive/realtime simulation environment shown in Fig. 1. We used two Novint Falcons to send position commands $\boldsymbol{p}_L^{\rm device}$ and $\boldsymbol{p}_R^{\rm device}$ without force feedback. For the reproducibility of the results, the experimenter moved the falcons by hands, the commands $\boldsymbol{p}_L^{\rm device}$ and $\boldsymbol{p}_R^{\rm device}$ were saved in data files, and the saved sequences of $\boldsymbol{p}_L^{\rm device}$ and $\boldsymbol{p}_R^{\rm device}$ were replayed in each scenario of the simulation. The contact forces between the robot and environment were simulated through a penalty-based frictional contact model proposed in [39], [40]. The timestep

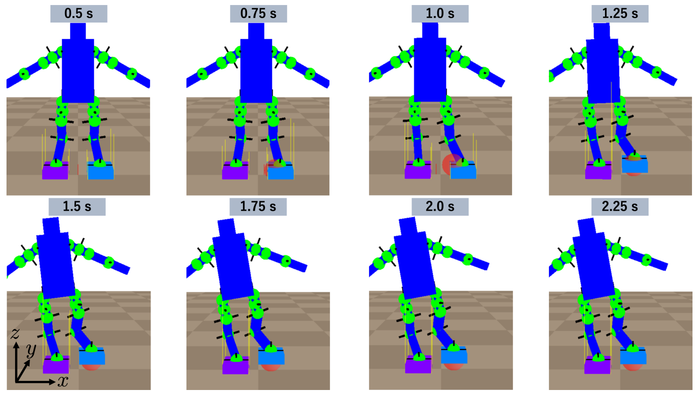

Fig. 9. Scenario 2: Snapshots of the simulation where a transition takes place from the double support phase to the single support phase through the proposed controller

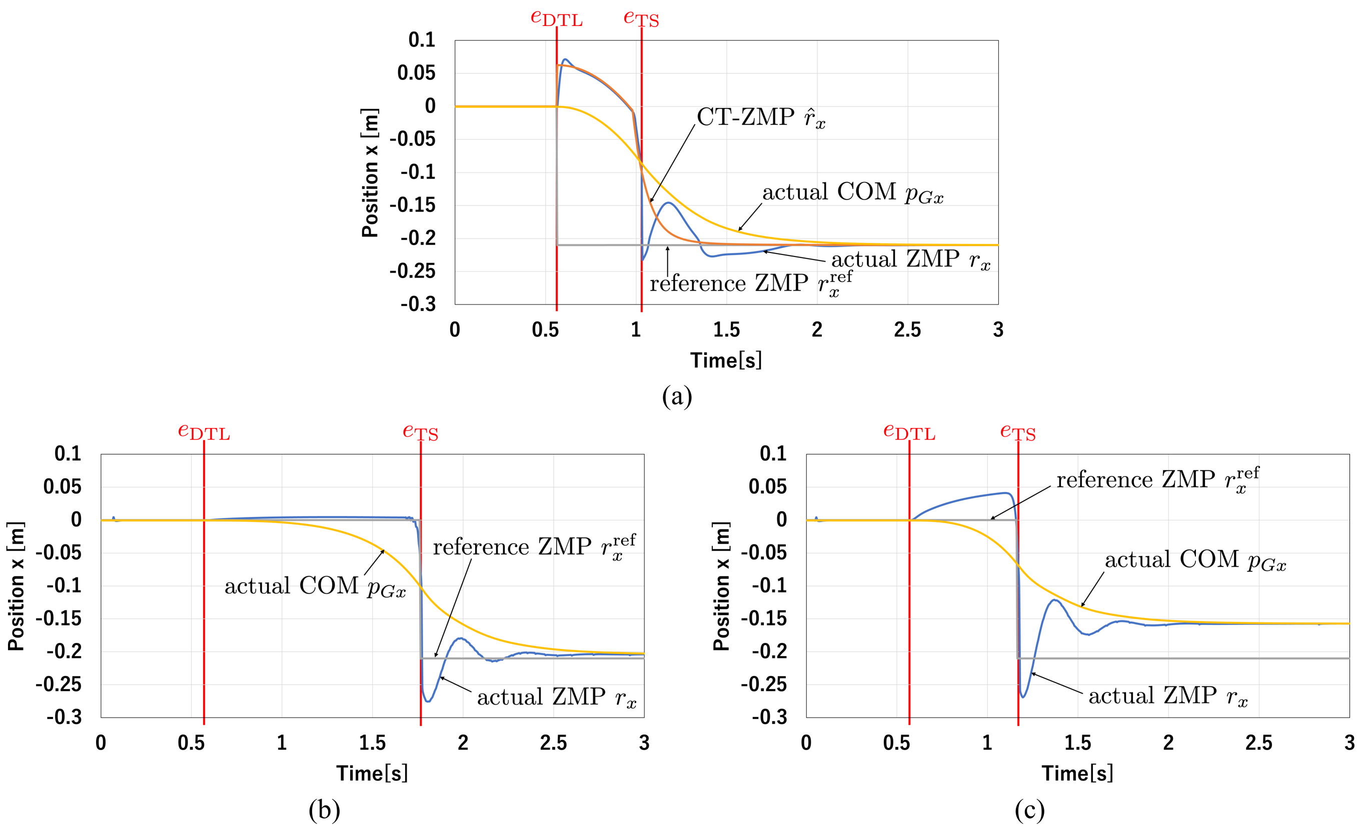

Fig. 10. Scenario 2: Simulation results of transition from the mode D to the mode SL with (a) the proposed controller and with the preview control with (b) $N = 240$ and (c) $N = 120$. The event e_{DTL} is made happen at $t = 0.565$ s by lifting the right haptic device. The event e_{TS} indicates the lifting of the right foot. The CT-ZMP $\hat{r}_x$ stands for the ZMP value calculated through (26)

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.

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.

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, white and gray), 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 haptic 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

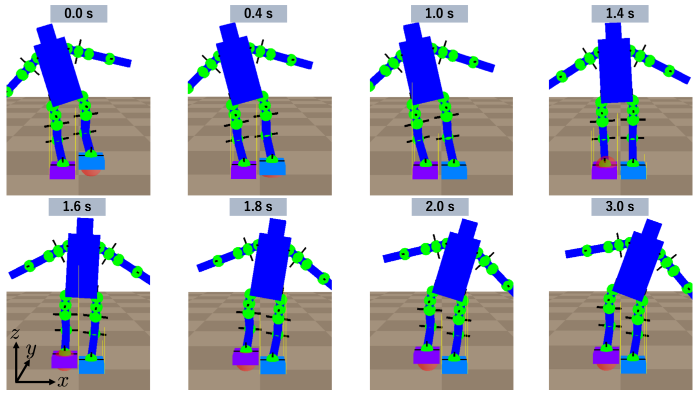

Fig. 11. Scenario 3: Snapshots of a simulation of mode transitions from SL via D and TR to SR

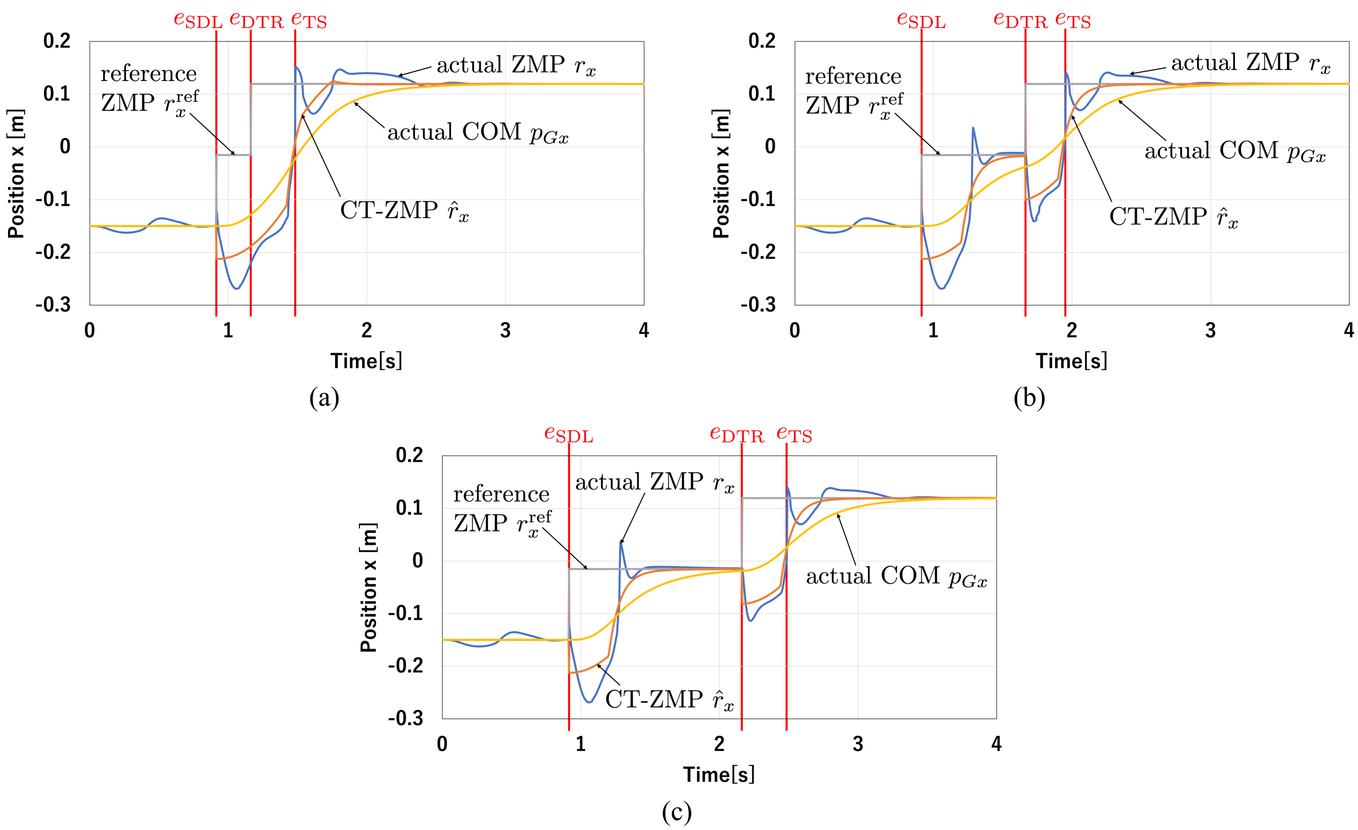

Fig. 12. Scenario 3: Simulation results of transitions from SL via D and TR to SR with different timing of the event e_{DTR}, which is the lifting of the left haptic device, at (a) $t \approx 1.2$ s, (b) 1.7 s, and (c) 2.2 s. The event e_{SDL} is the grounding of the right foot, which is initiated by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

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 operator to carefully manipulate the swing foot.

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 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 $[r_x^{\mathrm{ref}}, r_y^{\mathrm{ref}}]^T$ changes, the actual ZMP $[r_x, r_y]^T$ first moves in the opposite direction and later it converges to the reference ZMP $[r_x^{\mathrm{ref}}, r_y^{\mathrm{ref}}]^T$. 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.

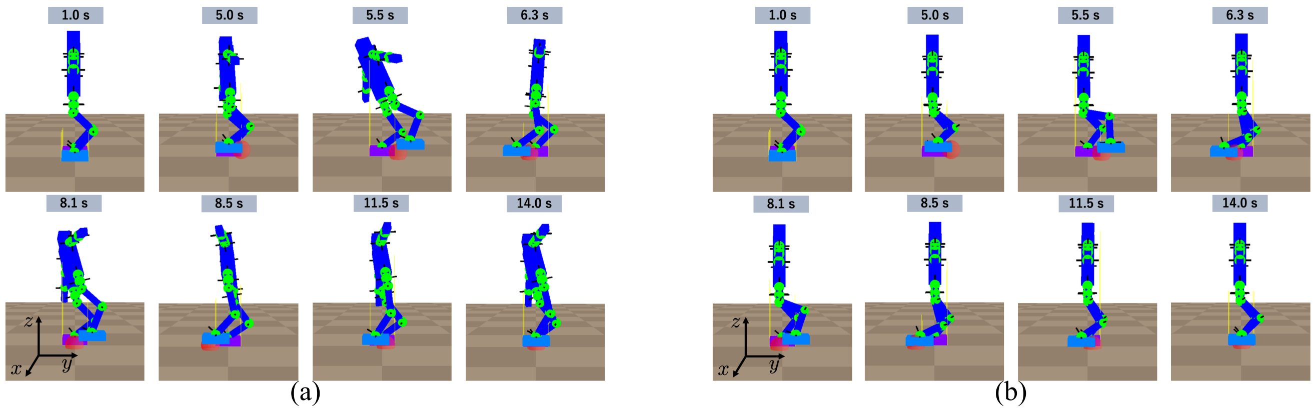

Fig. 13. Scenario 4: Fast swing of the leg in the single support phase with the body rotator (a) enabled and (b) disabled

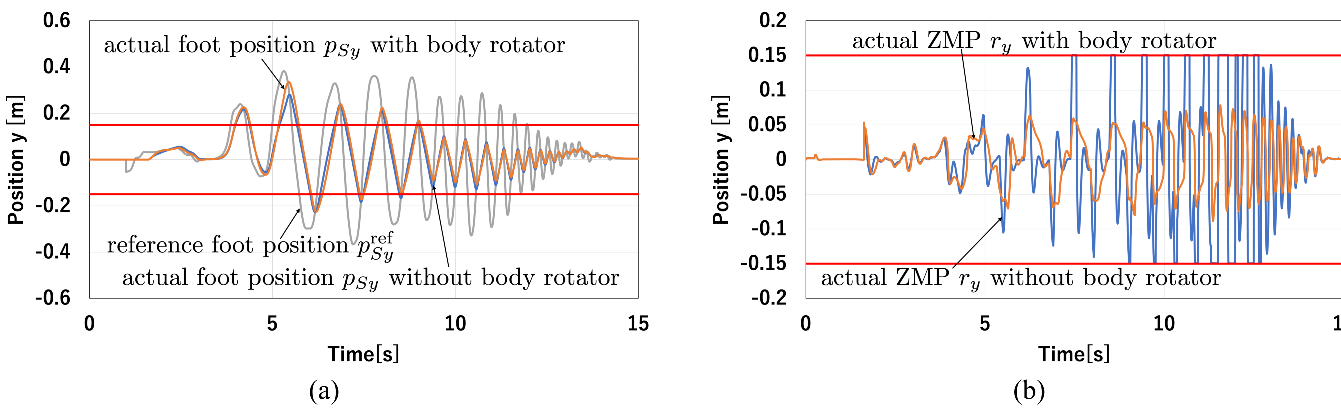

Fig. 14. Scenario 4: Simulation results of the proposed controller with the body rotator enabled and disabled; (a) The reference position p_{Sy}^{ref} and the actual position p_{Sy} of the foot in the y direction. (b) The actual ZMP r_y in the y direction. The red lines indicate the boundaries of the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

C. Scenario 2: Lifting one foot

In Scenario 2, transitions from the mode D to the mode SL were simulated to compare the proposed controller to Kajita et al.'s preview control [21] [20, Section 4.4]. The event e_{DTL} was made happen at $t = 0.565$ s by lifting the right haptic 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 [20] 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., [41], 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 haptic 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 haptic 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 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 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 the leg in the single support phase

In Scenario 4, we performed simulations of fast swing of the leg in the single-support mode SL. To show the effect of the body rotator, we compared the proposed controller to the one with the body rotator disabled, with which $\dot{\boldsymbol{L}}_{Gdxy}$ was set to be zero and $\boldsymbol{W}_A$ were 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. Conclusions

This paper has presented a controller for a step-by-step teleoperation scheme for humanoid robots, in which the user manipulates the foot positions of the robot at every step of walking. The main components of the controller, the COM shifter and the body rotator, are built upon a cart-flywheel-table model, which is a simplified dynamics model of a robot involving the angular momentum. 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 research should address the extension of the proposed controller to cope with external forces, more specifically, to be capable of pushing recovery motion [32], [42], [43] and automatic stepping motion [44], [45]. 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.

Acknowledgment

Our C++ source code for the simulator is adapted from Choreonoid [46], an open-source simulator developed by Dr. Shin'ichiro Nakaoka, the National Institute of Advanced Industrial Science and Technology (AIST), Japan.

Author contributions

Y.Z. conceived of the presented controller, performed the simulation experiments, and wrote the manuscript. R.K. reviewed and edited the manuscript and supervised the project.

Conflict of interest

The authors declare no potential conflict of interests.

References

[1] 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. (SII)*, 2014, pp. 590–596.

[2] 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, 2015.

[3] 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, 2016.

[4] 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.

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

[6] 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, 2019.

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

[8] 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)*, 2018, pp. 5835–5841.

[9] D. K. Prasanga, K. Tanida, K. Ohnishi, and T. Murakami, "Simultaneous bipedal locomotion based on haptics for teleoperation," *Adv. Robot.*, vol. 33, no. 15-16, pp. 824–839, 2019.

[10] 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, 2020.

[11] 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, 2021.

[12] J. Ramos and S. Kim, "Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation," *Science Robotics*, vol. 4, no. 35, p. eaav4282, 2019.

[13] 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)*, 2020, pp. 360–365.

[14] ——, "Reference ZMP generation for teleoperated bipedal robots walking on non-flat terrains," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, 2021, pp. 794–780.

[15] 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, no. 5-6, pp. 719–737, 2010.

[16] 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, 2009.

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

[18] 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, 2006.

[19] 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 Int. Conf. Humanoid Robots (Humanoids)*, 2015, pp. 936–940.

[20] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*, ser. Springer Tracts in Advanced Robotics. Springer, 2014, vol. 101.

[21] 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)*, 2003, pp. 1620–1627.

[22] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2006, pp. 137–142.

[23] 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, 2021.

[24] 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. (IROS)*, 2010, pp. 4489–4496.

[25] 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, 2014.

[26] S.-H. Lee and A. Goswami, "Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots," in *Proceedings 2007 IEEE international conference on robotics and automation*, 2007, pp. 4667–4672.

[27] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Auton. Robots*, vol. 33, no. 4, pp. 399–414, 2012.

[28] K. Yamamoto, T. Kamioka, and T. Sugihara, "Survey on model-based biped motion control for humanoid robots," *Adv. Robot.*, vol. 34, no. 21-22, pp. 1353–1369, 2020.

[29] T. Sugihara, "Standing stabilizability and stepping maneuver in planar bipedalism based on the best COM-ZMP regulator," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009, pp. 1966–1971.

[30] 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)*, 2019, pp. 1401–1406.

[31] 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)*, 2019, pp. 497–504.

[32] 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)*, 2006, pp. 200–207.

[33] 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)*, 2003, pp. 1644–1650.

[34] 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, 1987.

[35] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Trans. ASME: J. Dyn. Sys., Meas., Control.*, vol. 108, pp. 163–171, 1986.

[36] 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, 1995.

[37] 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, 2020.

[38] 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. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2011, pp. 13–18.

[39] 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, 2006.

[40] R. Kikuuwe and H. Fujimoto, "Incorporating geometric algorithms in impedance-and admittance-type haptic rendering," in *Proc. 2nd Joint EuroHap. Conf. Symp. Haptic Interfaces Virtual Environ., Teleoperator Syst.*, 2007, pp. 249–254.

[41] 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. (IROS)*, 2006, pp. 2993–2999.

[42] 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, 2022.

[43] 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, 2021.

[44] M. Morisawa, F. Kanehiro, K. Kaneko, N. Mansard, J. Sola, E. Yoshida, K. Yokoi, and J.-P. Laumond, "Combining suppression of the disturbance and reactive stepping for recovering balance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2010, pp. 3150–3156.

[45] T. Sugihara, "Reflexive step-out control superposed on standing stabilization of biped robots," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2012, pp. 741–746.

[46] S. Nakaoka, "Choreonoid: Extensible virtual robot environment built on an integrated GUI framework," in *PProc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, 2012, pp. 79–85.

APPENDIX A

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 $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ to its corresponding rotation vector (angle-axis representation), which have been used in, e.g., [20, 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 $\boldsymbol{a} \in \mathbb{R}^3$.