2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Torque-based Balancing Control for Teleoperated Bipedal Robots

Hanae Yamamoto[1], Katsuya Kanaoka[2], and Ryo Kikuuwe[1]

Abstract— **This paper proposes a torque-based leg controller to realize the teleoperated walking of biped robots. The controller is an extension of Kanaoka's balancing controller and allows the robot to switch legs and walk by leg force commands from the operator. It is made possible by switching the controller between the double- and single-support phases with appropriate mode transitions. The proposed controller was validated using a real-time simulator with haptic devices, in which the robot performed static and dynamic walking on a flat surface and slopes, following the operator's command through haptic devices.**

I. Introduction

Bipedal robots are potentially useful for working in unpredictable and unstructured environments, e.g., construction sites and disaster areas, on behalf of humans, but achieving full autonomy for these robots is still challenging. Teleoperation, which leverages human operators' intelligence, is a potential solution for such applications of bipedal robots [1]. Teleoperation is useful not only when the operator is in a remote site, but also when the operator is at close range or even onboard and the robot performs heavy-duty tasks that cannot be performed by humans. Because a bipedal robot is an intrinsically unstable mechanism with many degrees of freedom (DOFs), entrusting all DOFs to the human operator would make the robot prone to falling. Thus, teleoperated bipedal robots need to have a certain level of automatic or autonomous functions.

One imaginable scheme for teleoperated bipedal robots is to command the walking paths through Graphical User Interfaces (GUI) [2] or simple interface devices [3], [4]. Such methods should be combined with techniques for real-time generation of walking patterns [5], [6]. Another approach is to measure the operator's body motion through some sensor systems to replicate it with the robot. Such methods need to include appropriate automatic controllers to maintain the balance. Some studies [7], [8] focus on the dynamic aspects of the operator's body motion, which is translated into the robot's motion through an inverted pendulum model.

Robots working in rough terrains would need to allow the operator to manipulate the robot's leg motion in a step-by-step manner. Control schemes based on the dynamic motions, such as [7], [8], are not very suited for such a purpose, i.e., for precise positioning of each foot step. There are also quasi-static schemes, e.g., [9]–[14], which map the operator's motion to the robot's leg motion. One possible limitation of these schemes is that they are not suited for allowing the operator to manipulate leg forces. For example, the task of pushing a rock drill sideways against a rock wall requires lateral leg forces, which are able to not be produced by position-based schemes such as [10], [11].

Recently, Kanaoka [15] developed a torque-based balancing controller that can be combined with a teleoperation controller. More specifically, it can be combined with a force-projecting teleoperation controller. [16]–[18], which allows the operator to command forces to the robot and is suitable for force-aware tasks. Kanaoka's [15] controller is convenient to combine with the force-projecting teleoperation controller because they can be superposed with each other by simple summation of torque commands. This feature is not shared with other schemes such as the inverted pendulum-based schemes [7] and position-based schemes [10], [11]. The limitation of the controller is that the phase transitions, such as lifting and grounding the feet, were not carefully addressed in the controller.

This paper presents an extension of Kanaoka's [15] controller to achieve phase transitions at the operator's will, specifically, to allow the operator to lift and ground the robot's feet. This paper neglects the communication latency between the operator and the robot because our primary focus here is to realize heavy-duty robots that are either operated at close range or controlled directly by an onboard operator, where such latency is negligible. The controller was validated with an interactive/realtime simulator using two Novint Falcon devices.

This paper is organized as follows. Section II overviews Kanaoka's [15] controller framework on which the proposed method is based. Section III presents the proposed controller. Section IV details the interactive simulator environment with which the proposed controller was validated. Section V presents simulation results, and Section VI provides concluding remarks.

II. Basic framework of the controller

The controller proposed in this paper is based on the work by Kanaoka [15]. It consists of four main components, which are (a) the horizontal controller, (b) the center-of-pressure (COP) controller, (c) the vertical controller, and (d) the grounding controller, as illustrated in Fig. 1. The inputs to these controllers are the measured joint angles of the legs and the measured force and torque acting on the feet from the ground, which are measured by the load cells attached to the foot soles. The outputs of these controllers are summed up and sent to the actuators. In addition, this control framework is combined with the force-projecting bilateral control [16]–[18]; the force information from the leader robots, which are manipulated by the operator, is added to the torque command

[1]H. Yamamoto and R. Kikuuwe are with Hiroshima University, Japan.
[2]K. Kanaoka is with Man-Machine Synergy Effectors, Inc., Japan.

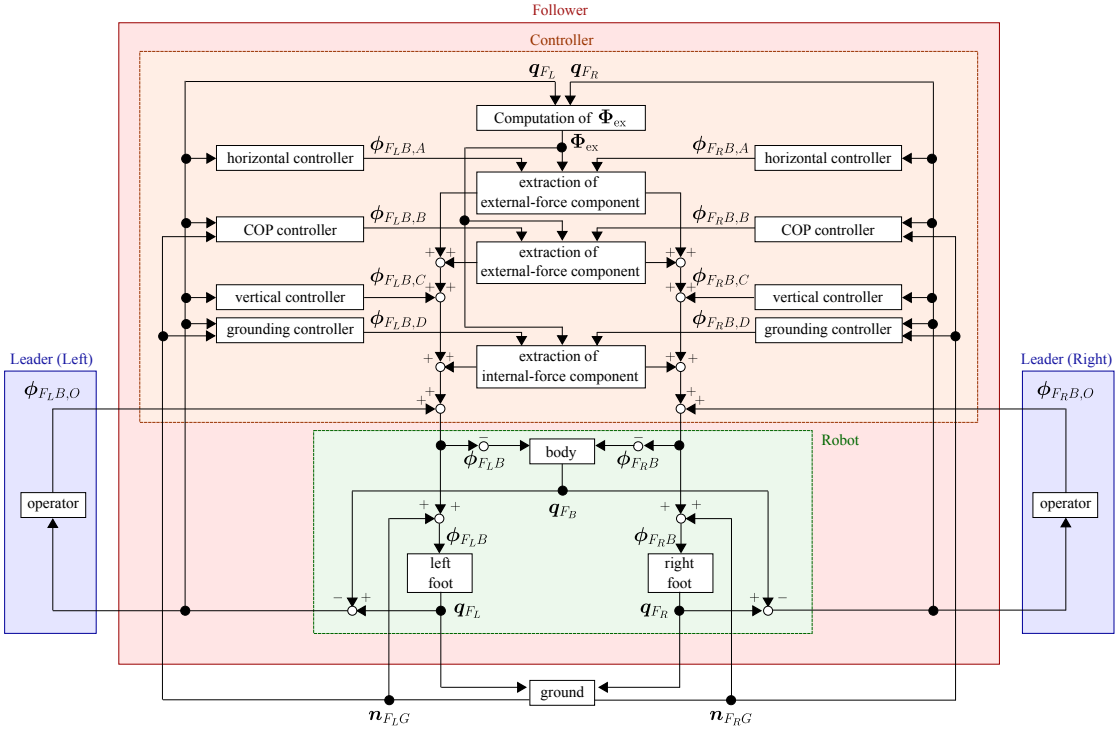


Fig. 1. Block diagram of the controller framework proposed by Kanaoka [15].

to the legs, and the foot positions relative to the body are sent to the leader robots. Again, this paper does not consider the communication latency between the leader robots and the follower robot.

In the following, B and F in left superscripts or right subscripts denote the coordinate frames attached to the body and either of the feet, respectively. For example, ${}^{B}\boldsymbol{R}_{F} \in \mathbb{R}^{3\times3}$ is the rotation matrix representing the attitude of a foot frame seen from the body frame. The vector ${}^{F}\boldsymbol{r}_{BF} \in \mathbb{R}^{3}$ denotes the vector from either foot to the body seen from the foot frame. The vector ${}^{F}\boldsymbol{\phi}_{BF} \in \mathbb{R}^{6}$ denotes a force-torque vector (the stack of a three-dimensional force vector and a three-dimensional torque vector) applied from either foot to the body seen from the foot frame. The vector ${}^{F}\boldsymbol{n}_{BF} \in \mathbb{R}^{3}$ denotes a torque vector in the same rule. We also use $\boldsymbol{o}_3 = [0,0,0]^T$ and $\boldsymbol{e}_z = [0,0,1]^T$.

A. Horizontal controller

The horizontal controller is a controller that maintains the body position relative to the feet in the directions parallel to each foot sole. It results in equilibrium when the body center is on the line perpendicular to the foot. By moving the body center horizontally toward the equilibrium point, the whole body can maintain balance. The output ${}^{B}\boldsymbol{\phi}_{FB,A}$ is given as follows:

$$ {}^{B}\boldsymbol{\phi}_{FB,A} = \begin{bmatrix} \boldsymbol{o}_3 \\ -{}^{B}\boldsymbol{R}_{F}\,{}^{F}\boldsymbol{n}_{BF,A} \end{bmatrix} \tag{1a} $$

$$ {}^{F}\boldsymbol{n}_{BF,A} = K_{P,A}\mathrm{ROT}(\boldsymbol{e}_z, {}^{F}\boldsymbol{r}_{BF}) - K_{B,A}\frac{{}^{F}\boldsymbol{r}_{BF}\times{}^{F}\dot{\boldsymbol{r}}_{BF}}{\|{}^{F}\boldsymbol{r}_{BF}\|^2} \tag{1b} $$

Here, K_P, K_B, K_I, and β are the proportional gain, derivative gain, integral gain, and the time constant, respectively, and $\mathrm{ROT}(\boldsymbol{q},\boldsymbol{p})$ is the rotational vector from $\boldsymbol{p}$ to $\boldsymbol{q}$, which is computed as follows:

$$ \begin{aligned} \mathrm{ROT}(\boldsymbol{q},\boldsymbol{p}) &\triangleq \mathrm{atan2}\left(\|\boldsymbol{p}\times\boldsymbol{q}\|, \boldsymbol{p}\cdot\boldsymbol{q}\right)\frac{\boldsymbol{p}\times\boldsymbol{q}}{\|\boldsymbol{p}\times\boldsymbol{q}\|} \\ &= \frac{\boldsymbol{p}\times\boldsymbol{q}}{\|\boldsymbol{p}\|\|\boldsymbol{q}\|\mathrm{sinc}\left(\mathrm{atan2}\left(\|\boldsymbol{p}\times\boldsymbol{q}\|, \boldsymbol{p}\cdot\boldsymbol{q}\right)\right)} \end{aligned} \tag{2} $$

B. COP controller

The COP controller is a controller to maintain the location of COP within the support polygon formed by the foot. It is also intended to maintain balance of the whole body by resisting external forces, in combination with the horizontal controller. The output ${}^{B}\boldsymbol{\phi}_{FB,B}$ is determined as follows [12]:

$$ {}^{B}\boldsymbol{\phi}_{FB,B} = \begin{bmatrix} \boldsymbol{o}_3 \\ -{}^{B}\boldsymbol{R}_{F}\,{}^{F}\boldsymbol{n}_{FGI,B} \end{bmatrix} \tag{3a} $$

$$ \mathcal{L}[{}^{F}\boldsymbol{n}_{FGI,B}] = \left(\frac{K_{I,B}}{s+1/\beta_B} + K_{P,B}\right)\mathcal{L}[{}^{F}\boldsymbol{n}_{FG}] \tag{3b} $$

Here, ${}^{F}\boldsymbol{n}_{FG}$ is the reaction torque from the ground to the foot seen from the foot frame. Equation (3b) can be viewed as a proportional-integral controller with a leaky integration.

C. Vertical controller

The vertical controller is a controller to maintain the height of the body. It applies a force to keep the vertical distance between the body and the foot. The output ${}^{B}\boldsymbol{\phi}_{FB,C}$ is given as follows [12]:

$$ {}^{B}\boldsymbol{\phi}_{FB,C} = \begin{bmatrix} K_{P,C}\boldsymbol{e}_z(H_d - \boldsymbol{e}_z^{T}{}^{B}\boldsymbol{r}_{FB}) - K_{B,C}{}^{B}\dot{\boldsymbol{r}}_{FB} \\ \boldsymbol{o}_3 \end{bmatrix} \tag{4} $$

where H_d is the target height of the body from the foot.

D. Grounding controller

The grounding controller is a controller to maintain the contact between a foot and the floor. As can be seen in Fig. 1, each leg possesses an instance of the controller, which works independently from each other. This controller acts only during the double support phase to prevent the feet from floating due to the COP controller. The output ${}^B\boldsymbol{\phi}_{FB,D}$ is given by the following equations [15].

$$ {}^B\boldsymbol{\phi}_{FB,D} = \begin{bmatrix} \boldsymbol{o}_3 \\ {}_B\boldsymbol{R}_F \, {}^F\boldsymbol{n}_{FGI,D} \end{bmatrix} \tag{5a} $$

$$ \mathcal{L}[{}^F\boldsymbol{n}_{FGI,D}] = \left(\frac{K_{I,D}}{s + 1/\beta_D} + K_{P,D} \right) \mathcal{L}[{}^F\boldsymbol{n}_{FG}] \tag{5b} $$

It should be noted that this controller is the sign-reversed version of the COP controller (3). The gains of this controller are set independently from those of the COP controller (3).

E. Extraction of External- and Internal-Force Components

In the double support phase, the forces generated by two legs act on one body, and they work to push the body in a direction and also work to compress or stretch the body. We can refer to the former and the latter as an external-force component and an internal-force component, respectively. Recall that the horizontal and COP controllers need to work on the balancing of the whole robot body, but the grounding controller needs to work on each foot. Thus, in the double support phase, we need to extract the external force component of the outputs of the horizontal and COP controllers, as well as the internal force component of the grounding controller.

Let the output of either of the horizontal, COP, or grounding controllers be denoted by $\boldsymbol{\phi} = [{}^B\boldsymbol{\phi}_{F_L B}{}^T, {}^B\boldsymbol{\phi}_{F_R B}{}^T]^T \in \mathbb{R}^{12}$. Then, its external- and internal-force components are, respectively, computed as follows:

$$ \boldsymbol{\phi}_{\mathrm{ex}} = \boldsymbol{\Phi}_{\mathrm{ex}} \boldsymbol{\phi} \tag{6a} $$

$$ \boldsymbol{\phi}_{\mathrm{in}} = (\boldsymbol{I} - \boldsymbol{\Phi}_{\mathrm{ex}}) \boldsymbol{\phi} \tag{6b} $$

where $\boldsymbol{\Phi}_{\mathrm{ex}} \in \mathbb{R}^{12\times 12}$ is an extraction matrix computed from Jacobian matrices and the rated torques of the actuators, detailed in [15].

III. Proposed Controller

The controller [15] overviewed in Section II is designed to maintain the robot's balance in each of the single and double support phases. It is insufficient to make transitions between the phases, which are indispensable for realizing walking. This section proposes an extension of the controller that facilitates transitions and allows the robot to walk.

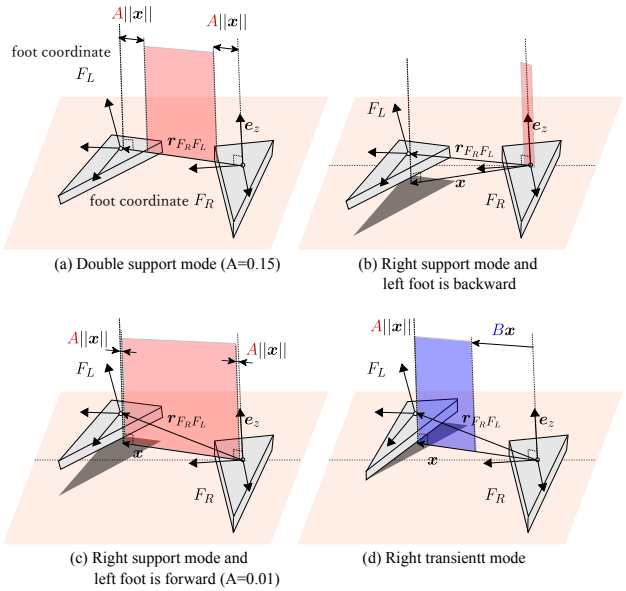
A. Modification of the Horizontal Controller

In Kanaoka's framework [15], the equilibrium point of the horizontal controller in the double support phase is located directly above each leg. It does not allow the horizontal movement of the COM of the body in the double support phase according to the operator's command. In the proposed modification, to allow the COM movement in the double support phase, the equilibrium point is located above the line segment between the two feet. To implement this modification, $\boldsymbol{e}_z$ in (1) is replaced by a target position ${}^F\boldsymbol{r}_{BF,d}$, which is determined by Algorithm 1. In Algorithm 1, $\boldsymbol{a}_z$ is the vertical unit vector seen from the foot frame. It depends on $\boldsymbol{R}_B$, which is the body attitude measured by IMU seen from the world frame. It allows for maintaining balance on tilted surfaces.

The variables $A \in [0, 0.5]$ and $B \in [0, 1]$ in Algorithm 1 are set according to the modes detailed in the next Section III-B. As illustrated in Fig. 2, the variables A and B

Algorithm 1 Computation of ${}^F\boldsymbol{r}_{BF,d}$

if supporting leg contacts at 3 points **then**
  $\boldsymbol{a}_z := \left({}^B\boldsymbol{R}_F\right)^T \left(\boldsymbol{R}_B\right)^T \boldsymbol{e}_z$
end if
if single support modes and events are not triggered **then**
  ${}^F\boldsymbol{r}_{BF,d} = \begin{cases} \boldsymbol{a}_z + K\boldsymbol{e}_y \, {}^F\boldsymbol{r}_{F_O F} & \text{if supporting leg and floating leg is forward} \\ \boldsymbol{a}_z & \text{if other supporting leg} \\ \boldsymbol{e}_z & \text{if floating leg} \end{cases}$
else
  $\boldsymbol{x} := (\boldsymbol{I}_3 - \boldsymbol{a}_z \boldsymbol{a}_z^T) \, {}^F\boldsymbol{r}_{F_O F}$
  $B := \min\left(1, \dfrac{\text{time since mode transition}}{T_B}\right)$
  if supporting leg in the transition modes **then**
    ${}^F\boldsymbol{r}_{BF,d} = \boldsymbol{a}_z(\boldsymbol{a}_z \cdot {}^F\boldsymbol{r}_{BF}) + \boldsymbol{x} \, \mathrm{gsat}\left(B, \dfrac{\boldsymbol{x} \cdot {}^F\boldsymbol{r}_{BF}}{\|\boldsymbol{x}\|^2}, 1 - A\right)$
  else if floating leg in the transition modes **then**
    ${}^F\boldsymbol{r}_{BF,d} = \boldsymbol{a}_z(\boldsymbol{a}_z \cdot {}^F\boldsymbol{r}_{BF}) + \boldsymbol{x} \, \mathrm{gsat}\left(A, \dfrac{\boldsymbol{x} \cdot {}^F\boldsymbol{r}_{BF}}{\|\boldsymbol{x}\|^2}, 1 - B\right)$
  else
    ${}^F\boldsymbol{r}_{BF,d} = \boldsymbol{a}_z(\boldsymbol{a}_z \cdot {}^F\boldsymbol{r}_{BF}) + \boldsymbol{x} \, \mathrm{gsat}\left(A, \dfrac{\boldsymbol{x} \cdot {}^F\boldsymbol{r}_{BF}}{\|\boldsymbol{x}\|^2}, 1 - A\right)$
  end if
end if



(a) Double support mode (A=0.15)
(b) Right support mode and left foot is backward
(c) Right support mode and left foot is forward (A=0.01)
(d) Right transientt mode

Fig. 2. Parameters A and B used in Algorithm 1.

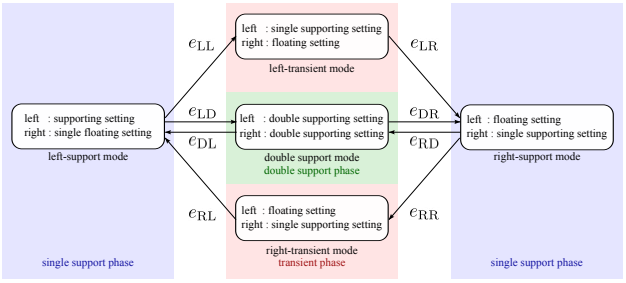
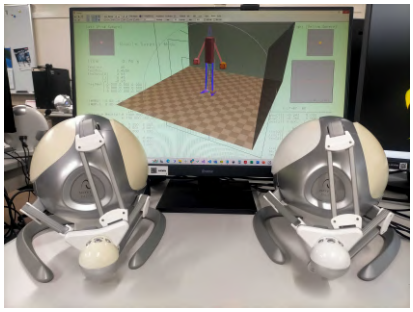
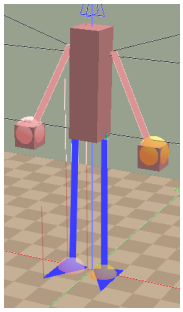


Fig. 3. Mode transition diagram.



(a) Falcons and simulator's screen



(b) Shape of the Robot

Fig. 4. Interactive simulation environment.

determine the region in which the COM can be in equilibrium with the horizontal controller, which varies according to the modes. With $A = 0$, the equilibrium can stay anywhere above the line segment connecting the two feet. With $A = 0.5$, the equilibrium is only above the midpoint of the two feet. In the transition modes, as illustrated in Fig. 2(d), the value B is gradually reduced to shrink the equilibrium range toward the next supporting leg.

B. Mode Transitions for Walking

The controller needs to be appropriately switched between the single- and double-support phases. To realize switching, we extend Kanaoka's framework by including a new mode-switching scheme. In the new scheme, each leg is under either the 'supporting setting' or the 'floating setting,' and the combinations of the settings of the legs are referred to as modes, as illustrated in Fig. 3. Each setting has a different set of parameter values for the controllers detailed in Section II.

The transitions among modes are performed as illustrated in Fig. 3, being triggered by the following events:

- e_{DL}: The COM of the torso has remained above the left foot for a certain time.
- e_{DR}: The COM of the torso has remained above the right foot for a certain time.
- e_{LD}: The left foot has been in contact with the ground for a certain time.
- e_{RD}: The right foot has been in contact with the ground for a certain time.
- e_{LL}: "The COM of the torso is not within the support polygon of the left leg" and "the number of ground contact points of the right foot is ≥ 1."
- e_{RR}: "The number of ground contact points of the left foot is ≥ 1" and "the COM of the torso is not within the support polygon of the right leg."
- e_{LR}: The COM of the torso is within the support polygon of the right leg.
- e_{RL}: The COM of the torso is within the support polygon of the left leg.

The operator is able to generate these events by manipulating their command force.

In the modes shown in Fig. 3, the transition from the left-support mode to the right-support mode may happen through the double-support mode or the left-transient mode. The former and the latter correspond to static walking and dynamic walking, respectively. Dynamic walking is realized when the transition is made from the left-support mode to the right-support mode without passing through the double-support mode. It achieves faster shifting of COM and faster walking than static walking, but it may increase the possibility of falling over. Therefore, if the robot is required to walk safely rather than fast, the controller should be set so that it ignores the events e_{LL}, e_{RR}, e_{RL}, and e_{LR}.

IV. Simulation

A. Interactive Simulator

We have developed an interactive simulator environment using the Microsoft Visual C++ language. Two Novint Falcon devices (hereafter referred to as "Falcon") from Novint Technologies were used as leader robots. A Falcon has three translational DOFs, which can be used to command the translational components of forces that should be generated by the legs.

Fig. 4(a) shows the display screen of the simulator and the Falcons. The robot in the simulator stands about 1.65 meters tall and weighs about 87 kilograms. Its legs are 0.35 meters long, and the surface area of the bottom of each foot is 0.034 square meters. The environment includes the floor (ground) and walls.

Fig. 4(b) shows the robot in the simulator. The robot consists of a torso, arms, hands, legs, and feet. The robot's torso and hands are modeled as rectangular prisms, while the feet are triangles. The bottom surface of the feet consists of a triangle with one inner point and two outer points. The spheres in the hands represent the centers of the hands, and those in the feet represent the positions of Falcons.

In the simulator, the torso, hands, and feet are modeled as independent rigid bodies, and the legs and arms are modeled as massless 6-DOF actuators that generate forces and torques, connecting the body to the legs or arms. The controller commands the forces and torques to be generated by the legs. The penalty-based frictional contact models proposed by Kikuuwe et al. [19] [20] are used for computing the contact forces.

Some parameter values for the controller are listed in Table I, Table II, and Table III. It should be noted that

TABLE I

DOUBLE SUPPORTING SETTING.

	horizontal	COP	vertical	grounding
$K_{P,*}$	1000.0 Nm	0.5	1000.0 Nm	0.5
$K_{B,*}$	150.0 Nms	-	150.0 Nms	-
$K_{I,*}$	-	0.01 s^{-1}	-	0.01 s^{-1}
β_*	-	1.0 s	-	1.0 s

TABLE II

SINGLE SUPPORTING SETTING.

	horizontal	COP	vertical	grounding
$K_{P,*}$	2000.0 Nm	0.5	1000.0 Nm	0.5
$K_{B,*}$	300.0 Nms	-	150.0 Nms	-
$K_{I,*}$	-	0.01 s^{-1}	-	0.01 s^{-1}
β_*	-	1.0 s	-	1.0 s

TABLE III

FLOATING SETTING.

	horizontal	COP	vertical	grounding
$K_{P,*}$	10.0 Nm	0.5	1000.0 Nm	0.5
$K_{B,*}$	1.5 Nms	-	150.0 Nms	-
$K_{I,*}$	-	0.01 s^{-1}	-	0.01 s^{-1}
β_*	-	1.0 s	-	1.0 s

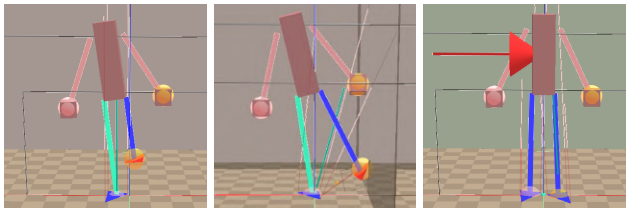
the proportional and derivative gains ($K_{P,A}$ and $K_{B,A}$) for the horizontal controller are common across the modes. The value of the ratio A was set to be 0.15 in the double support mode and 0.01 in the other modes, as shown in Fig. 2. The "certain time" in the definitions of the events e_{DL}, e_{DR}, e_{LD}, and e_{RD} in Section III-B was set to be 0.2 s.

We performed four scenarios of simulations. In scenarios 1 and 2, transitions to the transient phases were disabled to restrict the robot to the modes corresponding to static walking. In scenarios 3 and 4, all transitions shown in Fig. 3 were enabled.

B. Scenario 1: Maintaining Balance

The first set of simulations tested the balance maintenance capacity of the proposed controller in the single and double support phases. The results are shown in Fig. 5. In Fig. 5(a), the robot was in the single support phase, and an external force of 30 N was applied to the torso. In Fig. 5(b), the robot was operated to kick the wall on the right, and a sudden external force was applied to the robot through the right foot while the robot was on the left foot. In Fig. 5(c), an external force of 30 N was applied to the torso of the robot while the robot was in the double-support phase.

It is shown that the robot successfully maintained balance during both phases. The robot successfully maintained its balance even when it was pushing a wall with its foot and was subjected to a sudden external force. Additionally, the robot was able to maintain balance even when an external force of approximately 30 N was applied in the single and double support phases.



(a) Single support phase (b) Single support phase applied a sudden external force from a wall (c) Double support phase applied an external force

Fig. 5. Scenario 1: maintaining balance.

C. Scenario 2: Static Walking

Simulations of Scenario 2 were performed to test the static walking using the trigger events e_{DL}, e_{DR}, e_{LD}, and e_{RD}, excluding the events e_{LL}, e_{RR}, e_{RL}, and e_{LR}. That is, the transient modes (see Fig. 3) were not activated.

The results are shown in Fig. 6. The robot successfully walked the floor according to the operator's force commands from the interface devices. Such motions were made possible by the new horizontal controller, which allows for moving the COP in the double support phase from one foot to the other foot.

D. Scenario 3: Dynamic Walking

Scenario 3 was performed to test the dynamic walking using the trigger events e_{LL}, e_{RR}, e_{LR}, and e_{RL}. That is, the two transient modes (see Fig. 3) were also activated.

As shown in Fig. 7, the robot successfully performed dynamic walking according to the force commands from the operator through the interface devices. Currently, the success of dynamic walking depends on the operator's skill; the robot can fall over when it is operated by an unskilled operator. Further improvements of the algorithm, especially better guidelines for the parameter tuning, would be necessary to ensure successful dynamic walking even by unskilled operators.

E. Scenario 4: Walking on Slopes

Another set of simulations was performed as Scenario 4, in which the robot walked on a slope. It was mainly to test the effects of the improvement of the horizontal controller, in which the IMU information $\boldsymbol{R}_B$ is used to detect the body attitude. The robot successfully maintained balance during both the single and double support phases, both in the roll and pitch directions, as shown in Fig. 8.

The robot was also able to perform static walking on the slope. It moved forward, backward, and sideways on slopes tilted in both the roll and pitch directions, being operated by the operator through the interface device. The static walking was successful with the slope of 15 degrees, but the dynamic walking was not very successful with the slope of this angle.

V. CONCLUSION

This paper has proposed modifications to the balancing controller for bipedal robots proposed by Kanaoka [15]. The modifications are to allow for moving the COM in the double-support phase and making transitions between

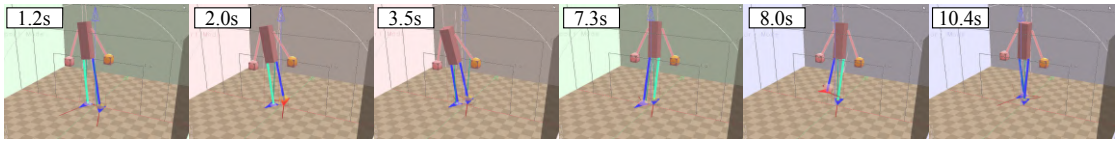


Fig. 6. Scenario 2: Snapshots of a simulation of static walking.

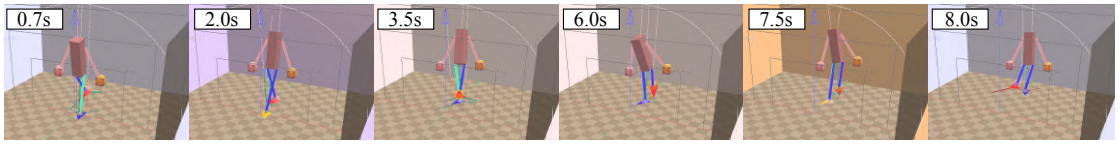
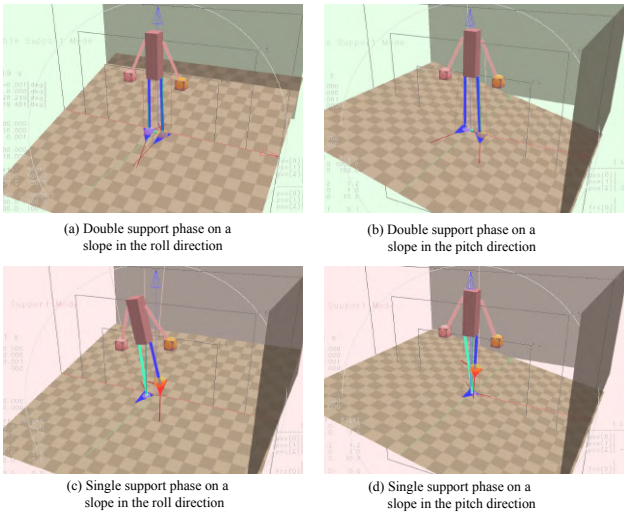


Fig. 7. Scenario 3: Snapshots of a simulation of dynamic walking.



(a) Double support phase on a slope in the roll direction

(b) Double support phase on a slope in the pitch direction

(c) Single support phase on a slope in the roll direction

(d) Single support phase on a slope in the pitch direction

Fig. 8. Scenario 4: walking on slopes.

the double-support phase and the single-support phase. The horizontal controller, which is a part of Kanaoka's controller, has been modified, and a mode transition strategy for the controller has been newly introduced. The controller was validated by an interactive simulator environment. In the simulator, the robot successfully performed both static and dynamic walking, and it was also able to walk on a slope.

Future work should focus on increasing the robot's walking speed and realizing successful dynamic walking regardless of the operator's skill. Additionally, developing the robot's walking capabilities on rough terrain will be essential. The improvement of the robots' balance maintenance capabilities under external forces should also be addressed.

REFERENCES

[1] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Trans. Robotics*, vol. 39, no. 3, pp. 1706–1727, 2023.

[2] 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. System Integration*, 2014, pp. 590–596.

[3] V. Moya, E. Slawiñski, V. Mut, D. Chávez, and B. Wagner, "Stable bilateral teleoperation control method for biped robots with time-varying delays," *J. Robotics*, vol. 2023, p. 3197743, 2023.

[4] M. Stilman, K. Nishiwaki, and S. Kagami, "Humanoid teleoperation for whole body manipulation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2008, pp. 3175–3180.

[5] J. Ding, M. Yang, J. Zhou, D. Yao, and X. Xiao, "Robust real-time walking pattern generation with dynamical consistency: an analytical method combined with optimal solution," in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, 2017, pp. 1806–1811.

[6] T. Sato, S. Sakaino, and K. Ohnishi, "Real-time walking trajectory generation method with three-mass models at constant body height for three-dimensional biped robots," *IEEE Trans. Industrial Electoronics*, vol. 58, no. 2, pp. 376–383, 2011.

[7] J. Ramos and S. Kim, "Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation," *Science Robotics*, vol. 4, p. eaav4282, 2019.

[8] G. Colin, J. Byrnes, Y. Sim, P. M. Wensing, and J. Ramos, "Whole-body dynamic telelocomotion: A step-to-step dynamics approach to human walking reference generation," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2023.

[9] I. Almetwally and M. Mallem, "Real-time tele-operation and tele-walking of humanoid robot Nao using Kinect depth camera," in *Proc. Int. Conf. Networking, Sensing and Control*, 2013, pp. 463–466.

[10] Y. Zhang and R. Kikuuwe, "COM shifter and body rotator for step-by-step teleoperation of bipedal robots," *IEEE Access*, vol. 11, pp. 25 786–25 800, 2023.

[11] ——, "Solving technical difficulties in implementing teleoperated biped walking on NAO robot," in *Proc. IEEE/SICE Int. Symp. System Integration*, 2024, pp. 1308–1313.

[12] 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 Robotics and Automation Letters*, vol. 5, no. 4, pp. 6419–6426, 2020.

[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. System Integration*, 2020, pp. 360–365.

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

[15] K. Kanaoka, "Legged robot," *Japanese Patent Application*, no. 2024-065892, 2024.

[16] ——, "Master/slave system and method for controlling same," *WIPO Patent Application*, no. WO 2011/115287 A1, 2011.

[17] ——, "Force projection type bilateral control for power-amplifying master-slave systems," in *Proc. Annual Conference of Robotics Society of Japan*, 2009, pp. 3C1–06, (in Japanese).

[18] R. Kikuuwe, K. Kanaoka, T. Kumon, and M. Yamamoto, "Phase-lead stabilization of force-projecting master-slave systems with a new sliding mode filter," *IEEE Trans. Control Systems Technology*, vol. 23, no. 6, pp. 2182–2194, 2015.

[19] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto, "Admittance and impedance representations of friction based on implicit Euler integration," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1176–1188, 2006.

[20] R. Kikuuwe and H. Fujimoto, "Incorporating geometric algorithms in impedance- and admittance-type haptic rendering," in *Proc. World Haptics Conf.*, 2007, pp. 249–254.