



Article

Balanced Standing on One Foot of Biped Robot Based on Three-Particle Model Predictive Control

Yong Yang ¹, Jiyuan Shi ¹, Songrui Huang ², Yuhong Ge ¹, Wenhan Cai ², Qingkai Li ², Xueying Chen ¹, Xiu Li ¹ and Mingguo Zhao ^{2,*}

¹ Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China

² Department of Automation, Tsinghua University, Beijing 100084, China

* Correspondence: mgzhao@tsinghua.edu.cn

Abstract: Balancing is a fundamental task in the motion control of bipedal robots. Compared to two-foot balancing, one-foot balancing introduces new challenges, such as a smaller supporting polygon and control difficulty coming from the kinematic coupling between the center of mass (CoM) and the swinging leg. Although nonlinear model predictive control (NMPC) may solve this problem, it is not feasible to implement it on the actual robot because of its large amount of calculation. This paper proposes the three-particle model predictive control (TP-MPC) approach. It combines with the hierarchical whole-body control (WBC) to solve the one-leg balancing problem in real time. The bipedal robot's torso and two legs are modeled as three separate particles without inertia. The TP-MPC generates feasible swing leg trajectories, followed by the WBC to adjust the robot's center of mass. Since the three-particle model is linear, the TP-MPC requires less computational cost, which implies real-time execution on an actual robot. The proposed method is verified in simulation. Simulation results show that our method can resist much larger external disturbance than the WBC-only control scheme.



Citation: Yong, Y.; Shi, J.; Huang, S.; Ge, Y.; Cai, W.; Li, Q.; Chen, X.; Li, X.; Zhao, M. Balanced Standing on One Foot of Biped Robot Based on Three-Particle Model Predictive Control. *Biomimetics* **2022**, *7*, 244. <https://doi.org/10.3390/biomimetics7040244>

Academic Editor: Xuechao Chen

Received: 29 November 2022

Accepted: 13 December 2022

Published: 16 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: model predictive control; whole-body control; biped robot balance

1. Introduction

1.1. Background

Motion control of a bipedal robot is a typical multi-task control problem. The robot has to keep its balance while moving or standing, interact with the environment, and complete tasks assigned by the user. Among the various tasks, balancing under external disturbance is a fundamental one for the robot to complete other tasks. Therefore, the balance control of a bipedal robot while walking or standing on both feet has been widely studied in recent years. However, the situation where robots stand on one leg and keep their balance has received little attention. Examples of such a situation include a humanoid trying to reach a distant object with its arms or a bipedal robot kicking a football. Compared with balancing on both feet, one-foot balancing presents new challenges, such as significantly reduced support polygons and reduced overall system stability. Furthermore, if the swing leg is heavy enough, its motion may significantly affect the motion of the entire robot [1]. In this sense, the problem of robot balance control with one foot is similar to that of robot balance control with two arms [2]. From a physical point of view, maintaining the balance of a biped robot with its arms reduces the difficulty of the control problem because the robot's supporting polygon remains unchanged, while extra degrees of freedom can be used to accomplish the balance task [3]. Therefore, the single-leg balancing stance method can be applied to dual-arm robots and vice versa.

1.2. Motivation

A bipedal robot's balancing on one foot is a typical multi-task control problem. The robot must keep the torso and the stance foot stable, move the swing foot properly, and

conform to external and internal constraints. The whole-body control (WBC) method has been widely adopted in the balancing control of bipedal robots due to its capabilities to generate dynamically feasible control outputs and to handle various tasks and constraints [4]. However, the WBC only considers the current states of the robot and cannot handle strongly infeasible reference trajectories [5]. The model predictive control (MPC) is commonly used in conjunction with WBC to complement these insufficiencies [6]. In case of large external disturbance, the MPC foresees the robot's future states according to a specific dynamical model and acts as an online re-planner for the WBC. However, the full dynamics of a one-foot standing robot are high degrees of freedom and strongly nonlinear. Therefore, an MPC employing full dynamics has to solve a complicated nonlinear optimization problem. Thus, the full-dynamics MPC is computationally expensive and cannot be executed in real time on an actual robot [7–11].

Current MPC algorithms for legged robots usually adopt simplified models [12], for example, the linear inverted pendulum (LIP) model [2,13,14], the single particle model [15] or the single rigid body (SRB) model [6], to balance between model precision and computation time. In the case of balancing on one foot, these models are unsuitable because they adopt the dynamics model of a particle or a rigid body. Thus, the motion of the swinging leg cannot be predicted in MPC. In this research, we propose to adopt a three-particle model in the model predictive control, which is called the three-particle model predictive control (TP-MPC) approach. It considers the motion of the swing leg and requires far less computation time than a full-dynamics MPC. As a result, the TP-MPC combined with the WBC will solve the problem of one-foot balance control for practical biped robots with real-time computing requirements.

1.3. Related Work

Simplified models have been widely used in the balance control of bipedal robots. For example, Vukobratovic et al. first proposed the zero moment point (ZMP) concept and applied it to balance a bipedal robot [16]. Next, Kajita et al. proposed the LIP model, which simplifies the biped robot into a single mass point moving horizontally and connected to the ground by a massless retractable stick [17]. Finally, Pratt et al. extended the LIP model to a Linear Flywheel Inverted Pendulum Model, which takes the robot's angular momentum into account [18]. Their method can resist a large external disturbance by computing the robot's Capture Point and Capture Region. Li et al. used the momentum to control the balance of a biped robot, and they found that linear momentum was more critical than angular momentum in the balance control task. When the two momentum tasks cannot be satisfied simultaneously due to a significant disturbance, the linear momentum will be given priority [19].

WBC has also been applied to the bipedal robot's balancing. Xie et al. used the hierarchical WBC to maintain a balance of a standing biped robot [20]. Their algorithm can execute on an onboard computer in real time by optimizing the computing process. However, their control scheme needs a planner to generate feasible trajectories followed by the WBC to resist large external disturbances. Kim et al. proposed a new whole-body control approach called whole-body motion control [21]. It can realize dynamic walking on a bipedal robot with a passive ankle and maintain balance under external disturbance.

Model predictive control repeatedly solves a finite-horizon optimal control problem starting from the system's current states [22]. Moreover, the MPC can prepare for future motions in advance. Therefore, it is very suitable for highly dynamic motion control of legged robots. Li et al. applied a single rigid body model predictive control to a biped robot with light legs [23]. The MPC generates optimal contact forces and torques at each foot, which are converted to joint torque commands through contact Jacobians. Their method enables the robot to walk at a speed of 1.6 m/s on complex terrains and achieve a wide range of dynamic motions. Luo et al. proposed a three-mass model predictive control method for gait control of biped robots [24]. Their planar three-mass model considers the robot's ZMP and angular momentum. The swing leg trajectories are generated by

heuristics rather than the MPC. Therefore, their method did not utilize the swing leg to balance the robot.

From a biomimicry point of view, swinging the legs is crucial for the balanced control of the torso [1]. Boston Dynamics released a video in 2013 of using swinging legs and arms to adjust posture but did not release relevant technical details [25]. To the best of our knowledge, other than the two works mentioned above, no literature has been published on the use of swinging legs to control the balance of biped robots.

1.4. Contribution

The main contributions of this paper are as follows:

1. The proposed TP-MPC method can generate feasible swing leg trajectories that balance the robot while standing on one foot. The WBC tracks the generated swing leg trajectories. As a result, the overall control scheme can resist large external disturbances.
2. The TP-MPC catches the main effects of the swing leg motion while being simple enough to operate at the same frequency as the WBC.

The overall structure of this paper is as follows: In Section 2, the three-particle model and the TP-MPC are derived. Section 3 introduces the hierarchical whole-body control approach and the overall control scheme. Section 4 discusses the simulation setups and results. Finally, Section 5 is for the conclusions and future work.

2. Three-Particle Model Predictive Control

In this section, we derive the three-particle model predictive control method. First, we introduce the simplified three-particle model for a bipedal robot standing on a single foot. Our MPC method requires to solve repeatedly a discrete, finite-horizon optimal control problem as follows:

$$\min_{\mathbf{X}, \mathbf{U}} = \sum_{k=0}^{n-1} \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1,ref}\|_{\mathbf{Q}_k} + \|\mathbf{u}_k\|_{\mathbf{R}_k} \quad (1)$$

$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k, \quad k = 0, 1, \dots, n-1 \quad (2)$$

$$\mathbf{l}b_k \leq \mathbf{C}_k \mathbf{u}_k \leq \mathbf{u}b_k, \quad k = 0, 1, \dots, n-1 \quad (3)$$

where n is the MPC's prediction length; \mathbf{x}_k and $\mathbf{x}_{k,ref}$ are the actual state and reference state at step k , \mathbf{u}_k is the input of step k , and \mathbf{Q}_k and \mathbf{R}_k are the weight matrices, $\|\lambda\|_{\Gamma} = \lambda^T \Gamma \lambda$ means the weighted value of λ of weight Γ . Equations (2) and (3) are equality and inequality constraints, respectively.

2.1. Three-Particle Simplified Model

Figure 1 shows the three-particle simplified model. The stance leg of the biped robot is fixed, and the position of the robot's center of mass can be adjusted by moving the torso and the swing leg to fall near the desired position. Here, we simplified the legs and the torso into three mass points without rotational inertia. In most cases, the torso's center of mass is located directly above the midpoint of the two hip joints but has an offset in the sagittal direction.

As shown in Figure 1, $\mathbf{p}_{st} = [p_{st,x} \ p_{st,y} \ p_{st,z}]^T$, $\mathbf{p}_{sw} = [p_{sw,x} \ p_{sw,y} \ p_{sw,z}]^T$ are the positions of the end of the stance leg and the swing leg in the world coordinate system, $\mathbf{p}_b = [p_{b,x} \ p_{b,y} \ p_{b,z}]^T$ is the position of the torso in the world coordinate system, $\dot{\mathbf{p}}_{st} = [\dot{p}_{st,x} \ \dot{p}_{st,y} \ \dot{p}_{st,z}]^T$, $\dot{\mathbf{p}}_{sw} = [\dot{p}_{sw,x} \ \dot{p}_{sw,y} \ \dot{p}_{sw,z}]^T$ are the velocities of the end of the stance leg and the swing leg in the world coordinate system, $\dot{\mathbf{p}}_b = [\dot{p}_{b,x} \ \dot{p}_{b,y} \ \dot{p}_{b,z}]^T$ is the velocity of the torso in the world coordinate system. m_{st} , m_{sw} , and m_b denote the mass of the stance leg, swing leg, and torso, respectively. In general, bipedal robots are symmetrical in the y direction, so the hip joint's distance from the hip's center in the y direction is l_y . Let the distance between the torso center of mass and the hip plane be l_z ,

and l_x is the distance between the torso center of mass in the x direction and the hip motor of both legs.

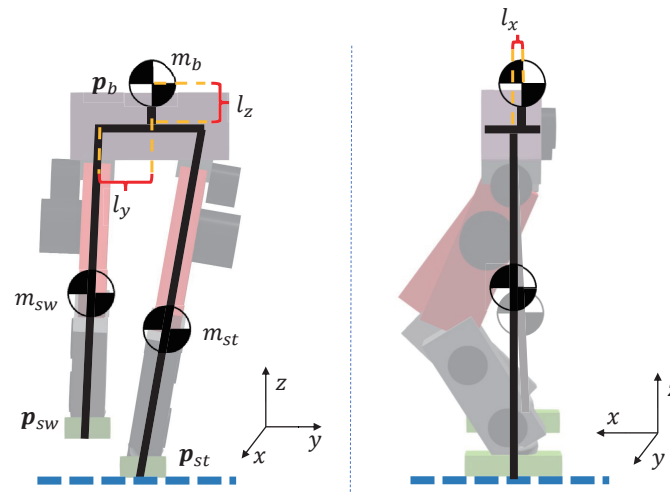


Figure 1. Three-Particle Model.

Let $p_{m,st}, p_{m,sw}$ be the center of mass position of the stance leg and the swing leg, respectively. To simplify the model, we assume that each leg's center of mass lies on the midpoint of the hip and foot, which means

$$p_{m,st} = \frac{1}{2} \begin{bmatrix} p_{b,x} + l_x + p_{st,x} \\ p_{b,y} + l_y + p_{st,y} \\ p_{b,z} - l_z + p_{st,z} \end{bmatrix} \quad (4)$$

$$p_{m,sw} = \frac{1}{2} \begin{bmatrix} p_{b,x} + l_x + p_{sw,x} \\ p_{b,y} - l_y + p_{sw,y} \\ p_{b,z} - l_z + p_{sw,z} \end{bmatrix} \quad (5)$$

The robot system's center of mass can be derived as follows:

$$m p_{CoM} = m_b p_b + m_{st} p_{m,st} + m_{sw} p_{m,sw} \quad (6)$$

where $m = m_b + m_{st} + m_{sw}$ represents the robot's total mass. Therefore, the robot's center of mass at time k is:

$$p_{CoM,k} = \frac{m_b p_{b,k} + m_{st} p_{m,st,k} + m_{sw} p_{m,sw,k}}{m} \quad (7)$$

Let $x = [p_b^\top \ p_{sw}^\top \ \dot{p}_b^\top \ \dot{p}_{sw}^\top]^\top$ be the state variables of the system, $u = [\ddot{p}_b^\top \ \ddot{p}_{sw}^\top]^\top$ be the control input. For the model predictive control design, using Taylor expansion, we obtain the state equation of the discrete-time linear system as:

$$x_{k+1} = \underbrace{\begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & \Delta t I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & \Delta t I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}}_A x_k + \underbrace{\begin{bmatrix} \frac{1}{2} \Delta t^2 I_{6 \times 6} \\ \Delta t I_{6 \times 6} \end{bmatrix}}_B u_k \quad (8)$$

Substituting Equations (4) and (5) into Equation (7) yields the location of the biped robot's center of mass:

$$p_{CoM,k} = \underbrace{\frac{1}{m} [m_1 I_{3 \times 3} \quad m_2 I_{3 \times 3} \quad 0_{3 \times 6}]}_C x_k + \underbrace{\frac{1}{2m} \begin{bmatrix} m_{st} x_{st} + (m_{sw} + m_{st}) l_x \\ m_{st} y_{st} + (m_{sw} - m_{st}) l_y \\ m_{st} z_{st} - (m_{sw} + m_{st}) l_z \end{bmatrix}}_D \quad (9)$$

where $m_1 = \frac{1}{2}(m_{sw} + m_{st}) + m_b$ and $m_2 = \frac{1}{2}m_{sw}$.

2.2. Tasks

For the one-foot balance control of a biped robot, the primary task is to minimize the position and velocity tracking errors of the robot's center of mass. The position tracking error of the swing leg should also be minimized. Moreover, the system input should be regularized to avoid excessive joint torques. Therefore, we formulate the cost function of the MPC as follows:

$$J = \min_u \sum_{k=1}^n \|p_{CoM,k} - p_{CoM,ref,k}\|_Q + \|\dot{p}_{CoM,k}\|_T + \|\bar{p}_{sw,k} - \bar{p}_{sw,ref,k}\|_S + \|\bar{p}_{b,k} - \bar{p}_{b,ref,k}\|_W + \|u_{k-1}\|_R \quad (10)$$

where $\|p_{CoM,k} - p_{CoM,ref,k}\|_Q$ is the CoM position tracking task where $p_{ref,k}$ is the reference trajectory of the CoM; $\|\dot{p}_{CoM,k}\|_T$ is the CoM velocity tracking task, where the desired velocity is set to 0; $\|\bar{p}_{sw,k} - \bar{p}_{sw,ref,k}\|_S$ is the position and velocity tracking task of the swing leg, where $\bar{p}_{sw,k}$ and $\bar{p}_{sw,ref,k}$ are the actual and desired trajectory of the swing leg, respectively; $\|\bar{p}_{b,k} - \bar{p}_{b,ref,k}\|_W$ is the position and velocity tracking task of the torso; $\bar{p}_{b,k}$ and $\bar{p}_{b,ref,k}$ are the actual trajectory and the desired trajectory of the torso; $\|u_k\|_R$ is the input penalty term; $Q \in \mathbb{R}^{3 \times 3}$, $T \in \mathbb{R}^{3 \times 3}$, $S \in \mathbb{R}^{6 \times 6}$, $W \in \mathbb{R}^{6 \times 6}$, $R \in \mathbb{R}^{6 \times 6}$ are the correspondent weight matrices.

Substituting Equation (8) into Equation (9), the position of the robot's center of mass can be expressed as:

$$p_{CoM,k} = Cx_k + D = CAx_{k-1} + CBu_{k-1} + D \quad (11)$$

The velocity of the center of mass can be obtained by taking the derivative of Equation (6). Then, its discrete form can be obtained:

$$\dot{p}_{CoM,k} = Fx_k = FAx_{k-1} + FBu_{k-1} \quad (12)$$

where

$$F = [0_{3 \times 3} \quad 0_{3 \times 3} \quad \frac{m_1}{m} I_{3 \times 3} \quad \frac{m_2}{m} I_{3 \times 3}] \quad (13)$$

The position and velocity of the swing leg can be derived by $\bar{p}_{sw,k} = Ex_k$, where

$$E = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (14)$$

is the selection matrix. Similarly, for the torso position and velocity, $\bar{p}_{b,k} = Gx_k$, where

$$G = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (15)$$

2.3. Constraints

In order to ensure the real-time performance of the model predictive control algorithm, there should not be too many constraints in the MPC. Here, we only constrain the length of the swing leg to avoid the kinematic singularity. Since the Cartesian distance $L = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$ is a nonlinear function of coordinates x, y, z , we use the Manhattan distance instead. The swing leg's length constraint is written as

$$|x_b - x_{sw}| + |y_b - y_{sw}| + |z_b - z_{sw}| \leq L_{leg} \quad (16)$$

Take the intermediate variable $s_{leg} = [s_x \ s_y \ s_z]^\top$, which satisfies

$$\begin{cases} s_{leg} \geq p_b - p_{sw} \\ s_{leg} \geq p_{sw} - p_b \end{cases} \quad (17)$$

Then, $|s_{leg}| \leq L_{leg}$ is equivalent to the constraints of Equation (16). The matrix form of the constraints can be derived as:

$$\begin{cases} s_{leg} \geq C_1 x_k \\ s_{leg} \geq C_2 x_k \\ \mathbf{1}^\top s_{leg} \leq L_{leg} \end{cases} \quad (18)$$

where

$$C_1 = [I_{3 \times 3} \quad -I_{3 \times 3} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 3}] \quad (19)$$

$$C_2 = [-I_{3 \times 3} \quad I_{3 \times 3} \quad \mathbf{0}_{3 \times 3} \quad \mathbf{0}_{3 \times 3}] \quad (20)$$

$$\mathbf{1} = [1 \ 1 \ 1]^\top \quad (21)$$

2.4. MPC Optimization Problem

2.4.1. Tasks

In order to improve the calculation efficiency, we eliminate the intermediate variables by substituting system dynamics into the cost function. Equation (8) can be expanded to

$$\begin{aligned} x_1 &= Ax_0 + Bu_0 \\ x_2 &= Ax_1 + Bu_1 = A^2x_0 + ABu_0 + Bu_1 \\ &\vdots \\ x_n &= Ax_{n-1} + Bu_{n-1} = A^n x_0 + A^{n-1}Bu_0 + \cdots + ABu_{n-2} + Bu_{n-1} \end{aligned} \quad (22)$$

Rewrite Equation (22) into matrix form as

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_X = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^n \end{bmatrix}}_{\tilde{A}} x_0 + \underbrace{\begin{bmatrix} B & \mathbf{0} & \cdots & \mathbf{0} \\ AB & B & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ A^{n-1}B & A^{n-2}B & \cdots & B \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix}}_U \quad (23)$$

Then, by replacing the matrices with X, \tilde{A}, \tilde{B} and U , Equation (23) can be rewritten as:

$$X = \tilde{A}x_0 + \tilde{B}U \quad (24)$$

Similarly, for the robot's center of mass position,

$$P_{CoM} = \tilde{C}X + \tilde{D} \quad (25)$$

where $\tilde{C} = \text{diag}(C, C, \dots, C) \in \mathbb{R}^{3n \times 12n}$, $\tilde{D} = [D^\top, D^\top, \dots, D^\top]^\top \in \mathbb{R}^{3n \times 1}$. The CoM velocity can be written as:

$$\dot{P}_{CoM} = \tilde{F}X \quad (26)$$

where

$$\tilde{F} = \text{diag}(F, F, \dots, F) \in \mathbb{R}^{3n \times 12n} \quad (27)$$

The position and velocity equations of the swing leg can be written as:

$$\bar{P}_{sw} = \tilde{E}X \quad (28)$$

where

$$\tilde{E} = \text{diag}(E, E, \dots, E) \in \mathbb{R}^{6n \times 12n} \quad (29)$$

Similarly, the position and velocity equations of the torso can be written as:

$$\bar{P}_b = \tilde{G}X \quad (30)$$

where

$$\tilde{G} = \text{diag}(G, G, \dots, G) \in \mathbb{R}^{6n \times 12n} \quad (31)$$

The stacked acceleration vector is as follows:

$$U = [u_0^\top, u_1^\top, \dots, u_{n-1}^\top]^\top \in \mathbb{R}^{6n \times 1} \quad (32)$$

Therefore, the cost function of Equation (10) can be rewritten as

$$\begin{aligned} \tilde{J} = \min_U & \|P_{CoM} - P_{CoM,ref}\|_{\tilde{Q}} + \|\dot{P}_{CoM}\|_{\tilde{T}} + \\ & \|\bar{P}_{sw} - \bar{P}_{sw,ref}\|_{\tilde{S}} + \|\bar{P}_b - \bar{P}_{b,ref}\|_{\tilde{W}} + \|U\|_{\tilde{R}} \end{aligned} \quad (33)$$

where

$$\tilde{Q} = \text{diag}(Q, Q \dots Q) \in \mathbb{R}^{3n \times 3n} \quad (34)$$

$$\tilde{T} = \text{diag}(T, T \dots T) \in \mathbb{R}^{3n \times 3n} \quad (35)$$

$$\tilde{S} = \text{diag}(S, S \dots S) \in \mathbb{R}^{6n \times 6n} \quad (36)$$

$$\tilde{W} = \text{diag}(W, W \dots W) \in \mathbb{R}^{6n \times 6n} \quad (37)$$

$$\tilde{R} = \text{diag}(R, R \dots R) \in \mathbb{R}^{6n \times 6n} \quad (38)$$

By eliminating the intermediate variables, Equation (33) can be rewritten as

$$\begin{aligned} \tilde{J} = \min_U & \|\tilde{C}\tilde{A}x_0 + \tilde{C}\tilde{B}U + \tilde{D} - P_{CoM,ref}\|_{\tilde{Q}} + \|\tilde{F}\tilde{A}x_0 + \tilde{F}\tilde{B}U\|_{\tilde{T}} + \\ & \|\tilde{E}\tilde{A}x_0 + \tilde{E}\tilde{B}U - \bar{P}_{sw,ref}\|_{\tilde{S}} + \|\tilde{G}\tilde{A}x_0 + \tilde{G}\tilde{B}U - \bar{P}_{b,ref}\|_{\tilde{W}} + \|U\|_{\tilde{R}} \end{aligned} \quad (39)$$

2.4.2. Constraints

According to Equation (3), the stacked form of the swing leg length constraint can be derived as

$$\begin{cases} \tilde{1}^\top \tilde{C}_1 \tilde{B}U \leq \tilde{L}_{leg} - \tilde{1}^\top \tilde{C}_1 \tilde{A}x_0 \\ \tilde{1}^\top \tilde{C}_2 \tilde{B}U \leq \tilde{L}_{leg} - \tilde{1}^\top \tilde{C}_2 \tilde{A}x_0 \end{cases} \quad (40)$$

where

$$\tilde{C}_1 = \text{diag}(C_1, C_1, \dots, C_1) \in \mathbb{R}^{3n \times 12n} \quad (41)$$

$$\tilde{C}_2 = \text{diag}(C_2, C_2, \dots, C_2) \in \mathbb{R}^{3n \times 12n} \quad (42)$$

$$\tilde{L}_{leg} = [L_{leg}, L_{leg}, \dots, L_{leg}]^\top \in \mathbb{R}^{n \times 1} \quad (43)$$

$$\tilde{1} = \text{diag}(1, 1, \dots, 1) \in \mathbb{R}^{3n \times n} \quad (44)$$

2.4.3. The Quadratic Programming Problem

The MPC problem can finally be transformed into the following quadratic optimization problem, which can be solved by off-the-shell QP solvers:

$$\begin{aligned} \min_{\mathbf{U}} \quad & \frac{1}{2} \mathbf{U}^\top \mathbf{H}_{qp} \mathbf{U} + \mathbf{g}_{qp}^\top \mathbf{U} \\ \text{s.t.} \quad & \mathbf{l}b_{qp} \leq \mathbf{A}_{qp} \mathbf{U} \leq \mathbf{u}b_{qp} \end{aligned} \quad (45)$$

where

$$\mathbf{H}_{qp} = \mathbf{H}_1 + \mathbf{H}_2 + \mathbf{H}_3 + \mathbf{H}_4 + \mathbf{H}_5 \quad (46)$$

$$\mathbf{g}_{qp} = \mathbf{g}_1 + \mathbf{g}_2 + \mathbf{g}_3 + \mathbf{g}_4 + \mathbf{g}_5 \quad (47)$$

$$\mathbf{A}_{qp} = \begin{bmatrix} \tilde{\mathbf{1}}^\top \tilde{\mathbf{C}}_1 \tilde{\mathbf{B}} \mathbf{U} \\ \tilde{\mathbf{1}}^\top \tilde{\mathbf{C}}_2 \tilde{\mathbf{B}} \mathbf{U} \end{bmatrix} \quad (48)$$

$$\mathbf{u}b_{qp} = \begin{bmatrix} \tilde{\mathbf{L}}_{leg} - \tilde{\mathbf{1}}^\top \tilde{\mathbf{C}}_1 \tilde{\mathbf{A}} \mathbf{x}_0 \\ \tilde{\mathbf{L}}_{leg} - \tilde{\mathbf{1}}^\top \tilde{\mathbf{C}}_2 \tilde{\mathbf{A}} \mathbf{x}_0 \end{bmatrix} \quad (49)$$

$$\mathbf{H}_1 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{C}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{C}} \tilde{\mathbf{B}} \quad (50)$$

$$\mathbf{H}_2 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{F}}^\top \tilde{\mathbf{T}} \tilde{\mathbf{F}} \tilde{\mathbf{B}} \quad (51)$$

$$\mathbf{H}_3 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{E}}^\top \tilde{\mathbf{S}} \tilde{\mathbf{E}} \tilde{\mathbf{B}} \quad (52)$$

$$\mathbf{H}_4 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{G}}^\top \tilde{\mathbf{W}} \tilde{\mathbf{G}} \tilde{\mathbf{B}} \quad (53)$$

$$\mathbf{H}_5 = 2\tilde{\mathbf{R}} \quad (54)$$

$$\mathbf{g}_1 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{C}}^\top \tilde{\mathbf{Q}} \left(-\mathbf{P}_{CoM,ref} + \tilde{\mathbf{C}} \tilde{\mathbf{A}} \mathbf{x}_0 + \tilde{\mathbf{D}} \right) \quad (55)$$

$$\mathbf{g}_2 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{F}}^\top \tilde{\mathbf{T}} \tilde{\mathbf{F}} \tilde{\mathbf{A}} \mathbf{x}_0 \quad (56)$$

$$\mathbf{g}_3 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{E}}^\top \tilde{\mathbf{S}} \left(\tilde{\mathbf{E}} \tilde{\mathbf{A}} \mathbf{x}_0 - \bar{\mathbf{P}}_{sw,ref} \right) \quad (57)$$

$$\mathbf{g}_4 = 2\tilde{\mathbf{B}}^\top \tilde{\mathbf{G}}^\top \tilde{\mathbf{W}} \left(\tilde{\mathbf{G}} \tilde{\mathbf{A}} \mathbf{x}_0 - \bar{\mathbf{P}}_{b,ref} \right) \quad (58)$$

$$\mathbf{g}_5 = \mathbf{0} \quad (59)$$

3. Hierarchical Whole-Body Control

3.1. Problem Formulation

The generalized coordinates of the floating base bipedal robot are defined as:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_f \\ \mathbf{q}_j \end{bmatrix} \in \mathbb{R}^{n_q} \quad (60)$$

where $\mathbf{q}_f \in \mathbb{R}^{n_f}$ represents the position and orientation of the robot's floating base relative to the inertial coordinate system, and $\mathbf{q}_j \in \mathbb{R}^{n_j}$ represents the angle of the actuated joints of the robot and $n_q = n_f + n_j$. The dynamic equation of the system is formulated as

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{0}_{n_f \times 1} \\ \boldsymbol{\tau}_j \end{bmatrix} + \mathbf{J}_c^\top(\mathbf{q}) \boldsymbol{\omega}_c \quad (61)$$

where $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the first and second-order derivative of \mathbf{q} , respectively, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n_q \times n_q}$ is the mass matrix of the robot, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n_q}$ collects the Coriolis force, centrifugal force and gravity, and $\boldsymbol{\tau}_j \in \mathbb{R}^{n_j}$ is the joint torque: $\boldsymbol{\omega}_{c,h} = \begin{bmatrix} \mathbf{f}_{c,h}^\top & \boldsymbol{\tau}_{c,h}^\top \end{bmatrix}^\top \in \mathbb{R}^{6 \times 1}$ is the external wrench applied to one leg, including three-dimensional external contact force and three-

dimensional contact torque. For the biped robot, $\omega_c = \begin{bmatrix} \omega_{c,0}^\top & \omega_{c,1}^\top \end{bmatrix}^\top \in \mathbb{R}^{12 \times 1}$. $J_c(q) \in \mathbb{R}^{12 \times n_q}$ is the contact Jacobian matrix.

In order to enable the robot to complete tasks with different priorities, a quadratic programming method with priority hierarchy was proposed [26]. The idea is to put the solution space of the upper priority level as a constraint of the lower level in the optimization problem. The optimization problem of the i th level task is defined as:

$$\begin{aligned} & \min_{\chi_i} \|\bar{A}_i \chi_i - \bar{b}_i\|_{Q_i} \\ \text{s.t. } & \bar{l}d_i \leq \bar{C}_i \chi_i \leq \bar{u}d_i \\ & \bar{A}_{i-1}^{aug} \chi_i = \bar{A}_{i-1}^{aug} \chi_{i-1}^* \\ & \bar{l}d_{i-1}^{aug} \leq \bar{C}_{i-1}^{aug} \chi_i \leq \bar{u}d_{i-1}^{aug} \end{aligned} \quad (62)$$

where \bar{A}_i and \bar{b}_i represent the task matrix and target vector corresponding to the current priority. If multiple tasks have the same priority, they could be weighted and combined by a diagonal matrix Q_i . \bar{C}_i , $\bar{l}d_i$ and $\bar{u}d_i$ represent the constraint matrix, lower bound, and upper bound of all constraints of the current priority. $\bar{A}_{i-1}^{aug} = \begin{bmatrix} \bar{A}_1^\top & \dots & \bar{A}_{i-1}^\top \end{bmatrix}^\top$ is the augmented task matrix corresponding to all tasks in the previous priority, and χ_{i-1}^* is the optimal solution of the previous priority. \bar{C}_{i-1}^{aug} , $\bar{l}d_{i-1}^{aug}$, $\bar{u}d_{i-1}^{aug}$, with a similar form of \bar{A}_{i-1}^{aug} , represent the augmented constraint matrix and bounds corresponding to all constraints in the previous priority. Here, the tasks in level i have higher priorities than tasks in level j when $i < j$. χ_i in level i represents the optimization variables to complete the tasks with priorities higher than $i + 1$, which means that the final optimal solution of multi-tasks with priority level n_l is $\chi_{n_l}^*$.

Here, we select the generalized joint acceleration and the contact wrench as the optimization variables in each level:

$$\mathbf{Y} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \omega_c \end{bmatrix} \quad (63)$$

3.2. Tasks

The definition of each level's tasks and constraints is given in Table 1. The floating base dynamics task is the basis of all motions of the mobile robot, and all the active forces come from joint motors, so the floating base dynamics task and joint moment constraints are set as the first priority. The linear momentum and trunk posture are critical factors for robot stability, while the ZMP constraint and the plantar friction cone constraint can ensure that the foot of the biped robot will not flip and slip, so these tasks and constraints are set as the second priority. The position of the robot's feet, posture task, and plantar force screw task are set as the third priority. The detailed definition of each task and constraint is as follows.

Table 1. Tasks, Constraints, and Priority Setting of Whole-Body Control for Single Leg Balance Standing.

Priority	Tasks	Tasks Dimension	Constraints	Constraints Dimension
1	Floating Base Dynamics	6	Joint Torque	10
2	Linear Momentum	6	ZMP	9
	Torso Posture		Friction Cone	
3	Foot Position & Posture	16		
	Contact Wrench			

3.2.1. Floating Base Dynamics Task

The first six rows of Equation (61) are part of the floating base dynamics, and it can be extracted by a selection matrix S_f :

$$S_f M(q) \ddot{q} + S_f h(q, \dot{q}) = S_f \begin{bmatrix} \mathbf{0}_{6 \times 1} \\ \boldsymbol{\tau}_j \end{bmatrix} + S_f J_c^\top(q) \boldsymbol{\omega}_c \quad (64)$$

where $S_f = \begin{bmatrix} \mathbf{I}_{n_f \times n_f} & \mathbf{0}_{n_f \times n_j} \end{bmatrix}$. Rearrange Equation (64) into the form of $\bar{A}_i \chi_i = \bar{b}_i$:

$$\begin{bmatrix} S_f M & -S_f J_c^\top \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \boldsymbol{\omega}_c \end{bmatrix} = -S_f h \quad (65)$$

Therefore, $\bar{A}_0 = \begin{bmatrix} S_f M & -S_f J_c^\top \end{bmatrix} \in \mathbb{R}^{n_f \times (n_q + 12)}$, $\bar{b}_0 = -S_f h \in \mathbb{R}^{n_f}$.

3.2.2. Centroidal Dynamics Task

Set the desired external force of the centroid task as $f_{G,des} \in \mathbb{R}^3$, and design the following PD controller to track the centroid task:

$$f_{G,des} = m[K_{p,0}(p_{G,tar} - p_G) + K_{d,0}(\dot{p}_{G,tar} - \dot{p}_G) + \ddot{p}_{G,tar}] \quad (66)$$

$$\bar{A}_{1,CoM} = \begin{bmatrix} S_G A_G & \mathbf{0}_{3 \times 12} \end{bmatrix} \in \mathbb{R}^{3 \times (n_q + 12)} \quad (67)$$

$$\bar{b}_{1,CoM} = f_{G,des} - S_G \dot{A}_G \dot{q} \in \mathbb{R}^3 \quad (68)$$

where $p_{G,tar} \in \mathbb{R}^3$, $\dot{p}_{G,tar} \in \mathbb{R}^3$ and $\ddot{p}_{G,tar} \in \mathbb{R}^3$ represent the desired position, velocity, and acceleration, respectively; $p_G \in \mathbb{R}^3$, $\dot{p}_G \in \mathbb{R}^3$ and $\ddot{p}_G \in \mathbb{R}^3$ represent the actual position, velocity, and acceleration, respectively; $S_G = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}$ represents the selected matrix, $A_G \in \mathbb{R}^{6 \times n_q}$ represents the centroidal momentum matrix [27]; $K_{p,0} \in \mathbb{R}^{3 \times 3}$ and $K_{d,0} \in \mathbb{R}^{3 \times 3}$ represent PD gain matrices.

3.2.3. Torso Orientation Task

Set the desired acceleration of the torso orientation task as $\ddot{\theta}_{t,des} \in \mathbb{R}^3$, and design the following PD controller to track the torso orientation:

$$\ddot{\theta}_{t,des} = K_{p1}(\theta_{t,tar} - \theta_t) + K_{d1}(\dot{\theta}_{t,tar} - \dot{\theta}_t) + \ddot{\theta}_{t,tar} \quad (69)$$

where $\theta_{t,tar} \in \mathbb{R}^3$, $\dot{\theta}_{t,tar} \in \mathbb{R}^3$, and $\ddot{\theta}_{t,tar} \in \mathbb{R}^3$ represent the desired torso orientation angle, angular velocity, and angular acceleration, respectively; $\theta_t \in \mathbb{R}^3$, $\dot{\theta}_t \in \mathbb{R}^3$ and $\ddot{\theta}_t \in \mathbb{R}^3$ represent the actual torso orientation angle, angular velocity, and angular acceleration, respectively; $K_{p,1} \in \mathbb{R}^{3 \times 3}$ and $K_{d,1} \in \mathbb{R}^{3 \times 3}$ represent PD parameter matrices.

The first six rows in the optimization variable \mathbf{Y} represent the position and orientation of the torso, and the fourth to sixth rows are the torso orientation, so, for the torso orientation task:

$$\bar{A}_{1,torso} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times (n_j + 12)} \end{bmatrix} \quad (70)$$

$$\bar{b}_{1,torso} = \ddot{\theta}_{t,des} \quad (71)$$

3.2.4. Feet Position and Orientation Task

As above, let $\ddot{r}_{f,des} \in \mathbb{R}^{12}$ be the desired acceleration of the foot posture, and the PD controller is designed as follows:

$$\ddot{r}_{f,des} = K_{p2}(r_{f,tar} - r_f) + K_{d2}(\dot{r}_{f,tar} - \dot{r}_f) + \ddot{r}_{f,tar} \quad (72)$$

where $\mathbf{r}_{f,tar} \in \mathbb{R}^{12}$, $\dot{\mathbf{r}}_{f,tar} \in \mathbb{R}^{12}$ and $\ddot{\mathbf{r}}_{f,tar} \in \mathbb{R}^{12}$ represent the desired foot posture, velocity, and acceleration, respectively; $\mathbf{r}_f \in \mathbb{R}^{12}$, $\dot{\mathbf{r}}_f \in \mathbb{R}^{12}$ and $\ddot{\mathbf{r}}_f \in \mathbb{R}^{12}$ represent the actual foot posture, velocity and acceleration, respectively; $\mathbf{K}_{p,2} \in \mathbb{R}^{12 \times 12}$ and $\mathbf{K}_{d,2} \in \mathbb{R}^{12 \times 12}$ represent PD parameter matrices.

The acceleration of the foot of the end can be obtained according to the following equation:

$$\mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \ddot{\mathbf{r}}_{f,des} \quad (73)$$

Then,

$$\overline{\mathbf{A}}_{2,feet} = [\mathbf{J}_c \quad \mathbf{0}_{12 \times 12}] \in \mathbb{R}^{12 \times (n_q + 12)} \quad (74)$$

$$\overline{\mathbf{b}}_{2,feet} = \ddot{\mathbf{r}}_{f,des} - \dot{\mathbf{J}}_c \dot{\mathbf{q}} \in \mathbb{R}^{12} \quad (75)$$

3.2.5. Contact Wrench Task

The contact wrench task tracks the desired contact wrench. Let $\boldsymbol{\omega}_{c,des}$ be the desired contact wrench. The last 12 dimensions of the optimization variable \mathbf{Y} are the contact wrench, so for the contact wrench task,

$$\overline{\mathbf{A}}_{2,wrench} = [\mathbf{0}_{12 \times n_q} \quad \mathbf{I}_{12 \times 12}] \quad (76)$$

$$\overline{\mathbf{b}}_{2,wrench} = \boldsymbol{\omega}_{c,des} \in \mathbb{R}^{12} \quad (77)$$

3.3. Constraints

3.3.1. Joint Torque Constraint

The last ten rows of Equation (61) represent the actuated joints, which should be limited by torque:

$$\mathbf{S}_j \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{S}_j \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}_j \begin{bmatrix} \mathbf{0}_{6 \times 1} \\ \boldsymbol{\tau}_j \end{bmatrix} + \mathbf{S}_j \mathbf{J}_c^\top(\mathbf{q}) \boldsymbol{\omega}_c \quad (78)$$

where $\mathbf{S}_j = [\mathbf{0}_{n_j \times n_f} \quad \mathbf{I}_{n_j \times n_j}]$ represents the selection matrix. The joint torque can be derived as:

$$\boldsymbol{\tau}_j = \mathbf{S}_j \mathbf{M} \ddot{\mathbf{q}} + \mathbf{S}_j \mathbf{h} - \mathbf{S}_j \mathbf{J}_c^\top \boldsymbol{\omega}_c \quad (79)$$

Therefore, the joint torque constraint is:

$$-\boldsymbol{\tau}_{limit} \leq \boldsymbol{\tau}_j \leq \boldsymbol{\tau}_{limit} \quad (80)$$

where $\boldsymbol{\tau}_{limit} \in \mathbb{R}^{n_j}$ is the torque limit of each actuated joint. Then, it can be obtained that:

$$\overline{\mathbf{C}}_0 = [\mathbf{S}_j \mathbf{M} \quad -\mathbf{S}_j \mathbf{J}_c^\top] \in \mathbb{R}^{n_j \times (n_q + 12)} \quad (81)$$

$$\overline{\mathbf{ld}}_0 = -\boldsymbol{\tau}_{limit} - \mathbf{S}_j \mathbf{h} \in \mathbb{R}^{n_j} \quad (82)$$

$$\overline{\mathbf{ud}}_0 = \boldsymbol{\tau}_{limit} - \mathbf{S}_j \mathbf{h} \in \mathbb{R}^{n_j} \quad (83)$$

3.3.2. ZMP Constraint

According to [28,29], the ZMP constraint on x direction can be written as

$$l_x^- \leq \frac{-\tau_{c,y} - f_{c,x}d}{f_{c,z}} \leq l_x^+ \quad (84)$$

where l_x^- and l_x^+ are the minimum and maximum values of ZMP, d is the vertical distance between the foot joint and the ground. $\tau_{c,y}$ is the moment applied to the foot in the direction of y , $f_{c,x}$, and $f_{c,z}$ represent the force applied to the foot in the direction of x and z . For the y direction, the ZMP constraint can be derived similarly. Rewrite the ZMP constraints in matrix form, where

$$C_{zmp} = \begin{bmatrix} -d & 0 & -l_x^- & 0 & -1 & 0 \\ d & 0 & l_x^+ & 0 & 1 & 0 \\ 0 & -d & -l_y^- & -1 & 0 & 0 \\ 0 & d & l_y^+ & 1 & 0 & 0 \end{bmatrix} \quad (85)$$

$$\bar{C}_{1,zmp} = \begin{bmatrix} \mathbf{0}_{4 \times 16} & C_{zmp} & \mathbf{0}_{4 \times 6} \\ \mathbf{0}_{4 \times 16} & \mathbf{0}_{4 \times 6} & C_{zmp} \end{bmatrix} \quad (86)$$

$$\bar{ld}_{1,zmp} = \mathbf{0}_{8 \times 1} \quad (87)$$

$$\bar{ud}_{1,zmp} = \infty_{8 \times 1} \quad (88)$$

3.3.3. Foot Friction Cone Constraint

The stance leg's ground reaction force should stay in the friction cone to avoid slipping. Here, we employ a pyramid approximation of the friction cone and formulate the foot friction cone constraint as

$$|f_{x,y}| \leq \mu f_z \quad (89)$$

where μ is the coefficient of friction; $f_{x,y}$ is the resultant force in the direction of x and y , and f_z is the force in the direction of z . Expand and arrange the aforementioned equation into the matrix form:

$$C_f = \begin{bmatrix} 1 & 0 & \mu \\ -1 & 0 & \mu \\ 0 & 1 & \mu \\ 0 & -1 & \mu \\ 0 & 0 & 1 \end{bmatrix} \quad (90)$$

$$\bar{C}_{1,friction} = \begin{bmatrix} \mathbf{0}_{5 \times 16} & C_f & \mathbf{0}_{5 \times 3} & \mathbf{0}_{5 \times 3} & \mathbf{0}_{5 \times 3} \\ \mathbf{0}_{5 \times 16} & \mathbf{0}_{5 \times 3} & \mathbf{0}_{5 \times 3} & C_f & \mathbf{0}_{5 \times 3} \end{bmatrix} \quad (91)$$

$$\bar{lb}_{1,friction} = \mathbf{0}_{10 \times 1} \quad (92)$$

$$\bar{ub}_{1,friction} = \infty_{10 \times 1} \quad (93)$$

3.4. Control Framework

Figure 2 shows the overall control framework. Both TP-MPC and WBC run at 1 kHz. The input of TP-MPC is the desired position of CoM, swing leg, and torso in the future n steps. WBC then takes the target trajectory of the swing leg calculated by TP-MPC and the target trajectory of CoM and torso orientation given by the stand planner as input. Although TP-MPC can also obtain the trajectories of the torso and CoM, we do not use them in our controller.

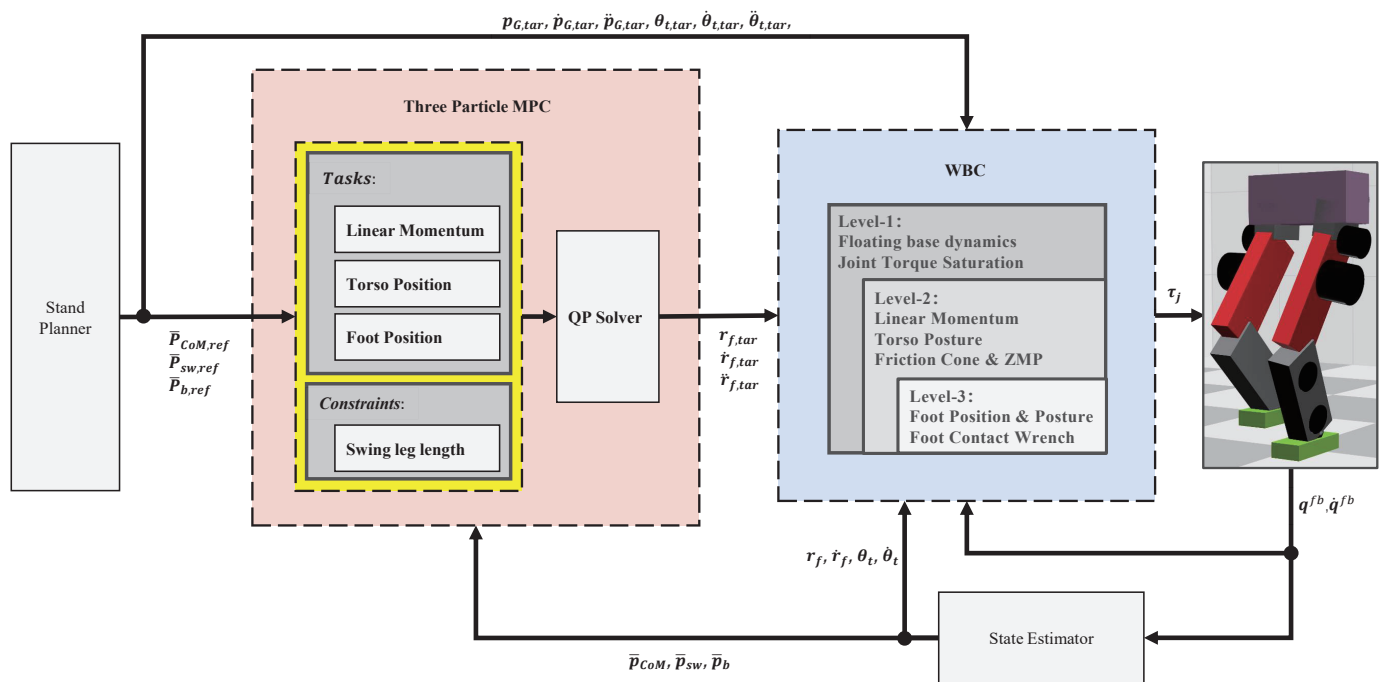


Figure 2. Overall control framework.

4. Simulation Results and Discussion

4.1. Simulation Setup

The simulation platform is the open-source robot simulator Webots [30]. The control algorithms are implemented with C++. We adopt the rigid body dynamics library (RBDL) [31] to calculate robot kinematics and dynamics and the qpOASES [32] toolkit to solve quadratic programming problems. The simulation environment and the controller run on a computer with an i7-11800H 2.30 GHz processor, 32 GB memory, and RTX3070 8 GB RAM GPU.

The bipedal robot in simulation has ten actuated joints and a total weight of 45 kg. The weights of the torso and a single leg are 22 kg and 11.5 kg, respectively. A single leg accounts for around one-fourth of the robot's total weight, which implies a significant effect of the swing leg's motion on the entire robot states. The robot's soles are 0.22 m-by-0.1 m rectangles.

As shown in Figure 3, the biped robot stands on a single leg, and a rigid hanging ball swings down to hit the robot. The suspension point of the ball is vertically above the robot. The ball is released from the same height and angle with zero initial speed. The robot's torso is hit 0.8 m above the ground in the frontal or the side direction. The initial height of the robot's CoM is 0.604 m. Therefore, the hit will exert a net torque concerning the robot's CoM. By increasing the rigid ball's mass, the impulse received by the robot at the instance of impact increases, meaning larger external disturbance. We compare our control scheme with the WBC-only control scheme in each experiment. For both control schemes, the WBC part is the same. For the WBC-only scheme, the desired torso and leg trajectories remain equal to their initial values.

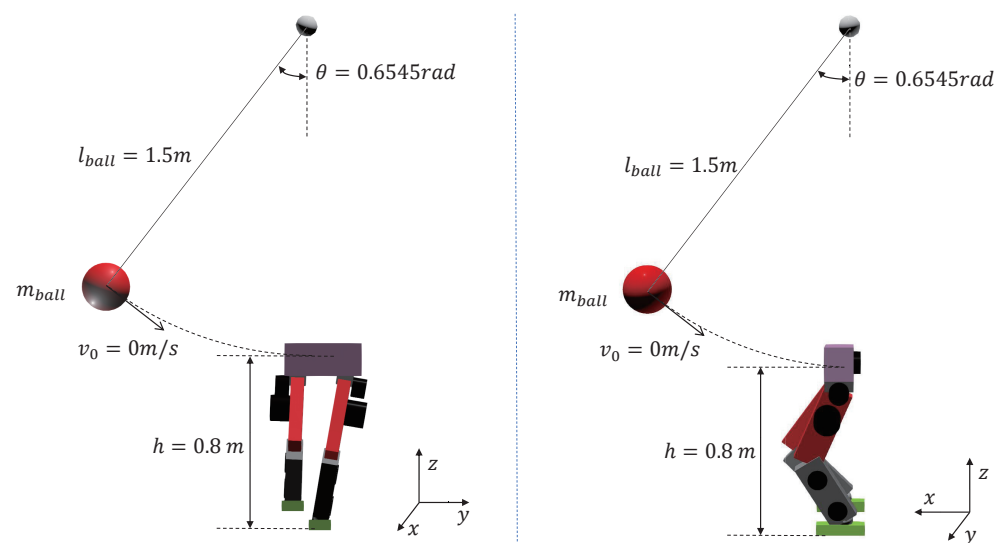


Figure 3. Schematic diagram of collision experiment.

4.2. Results

4.2.1. Frontal Impact

In the frontal impact experiment, the rigid ball swings along the x -axis and hits the center of the torso's front surface. When the mass of the rigid ball is 9 kg or 11 kg, the robot's CoM trajectories during the impact and recovery stages are shown in Figures 4 and 5, respectively. In the 9 kg case, both control schemes can resist the external disturbance and bring the robot's CoM back to its initial position, but our MPC-WBC scheme achieves smaller CoM motions and faster recovery than the WBC-only scheme. For both control schemes, Figures 6 and 7 show the actual trajectories of the swing leg's foot end and the correspondent reference trajectories given to WBC. For the WBC-only scheme, the WBC's reference trajectories are constant, while the actual trajectories deviate significantly from the reference. This deviation results from the WBC sacrificing the foot-end tracking task for the high-priority CoM tracking task. For our MPC-WBC scheme, the reference foot end trajectories provided by the MPC and the actual trajectories almost coincide with each other. This suggests that the MPC generates swing foot trajectories that are coherent with the CoM tracking targets. It is also observed that the MPC-WBC scheme generates larger swing foot motions than the WBC-only scheme, which confirms the swing leg's vital role in balancing the robot.

When the ball's mass increases to 11 kg, the WBC-only scheme causes the divergence of the robot's CoM trajectories and thus the failure of the entire robot. However, our MPC-WBC scheme can still withstand the impact. Screenshots in Figure 8 depict the robot's motions during the impact and recovery process under our MPC-WBC scheme. Significant forward and upward motions of the swing leg can be observed after the impact, implying that our method successfully utilizes the swing leg motions to balance the robot.

Figures 9 and 10 show the robot's centroidal pitch angular momentum. Although the rotational dynamics are not included in our TP-MPC, the simulation results show that our method is sufficient to deal with a certain degree of torque perturbation relative to the CoM.

Figure 11 shows the computation time cost of the TP-MPC algorithm. It can be seen that most of the computations are within 500 μ s.

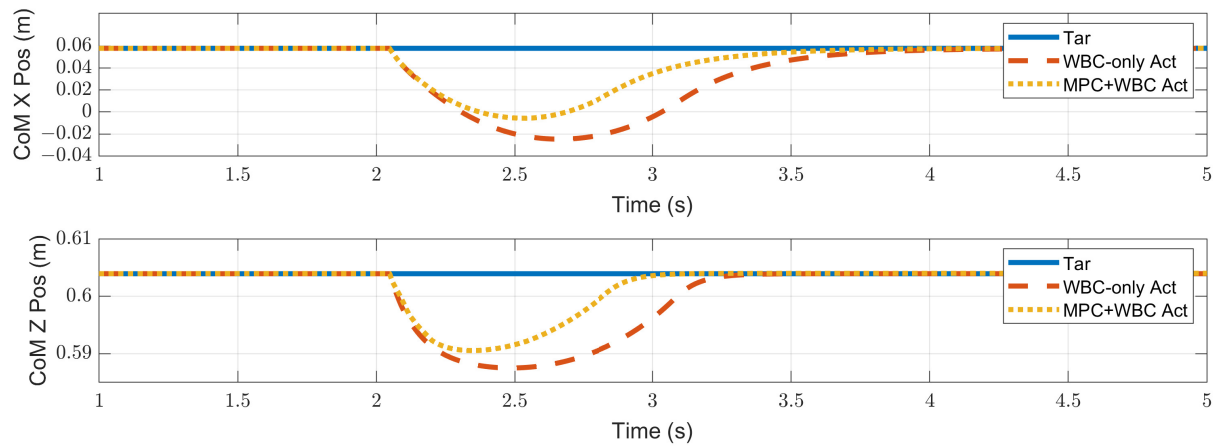


Figure 4. CoM position of the robot under the frontal impact of a 9 kg rigid ball.

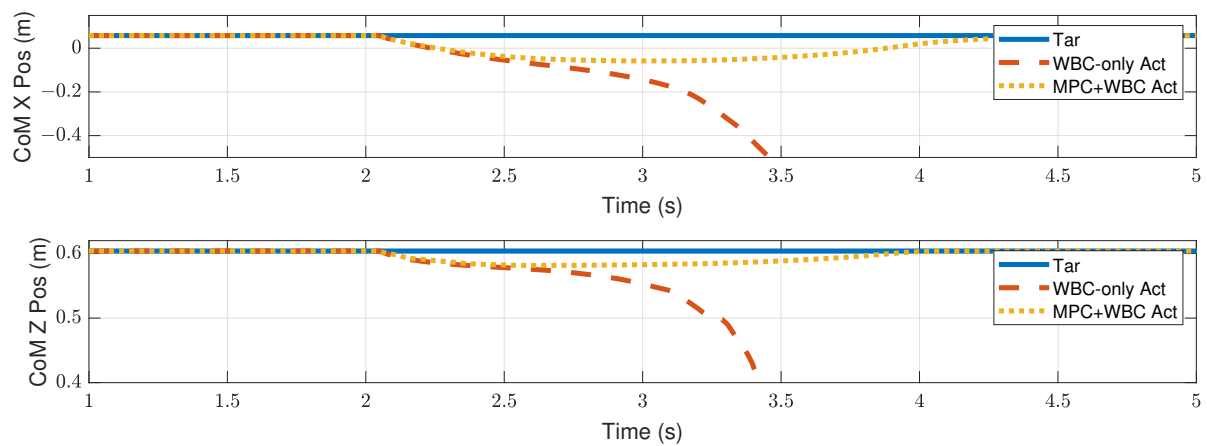


Figure 5. CoM position of the robot under the frontal impact of an 11 kg rigid ball.

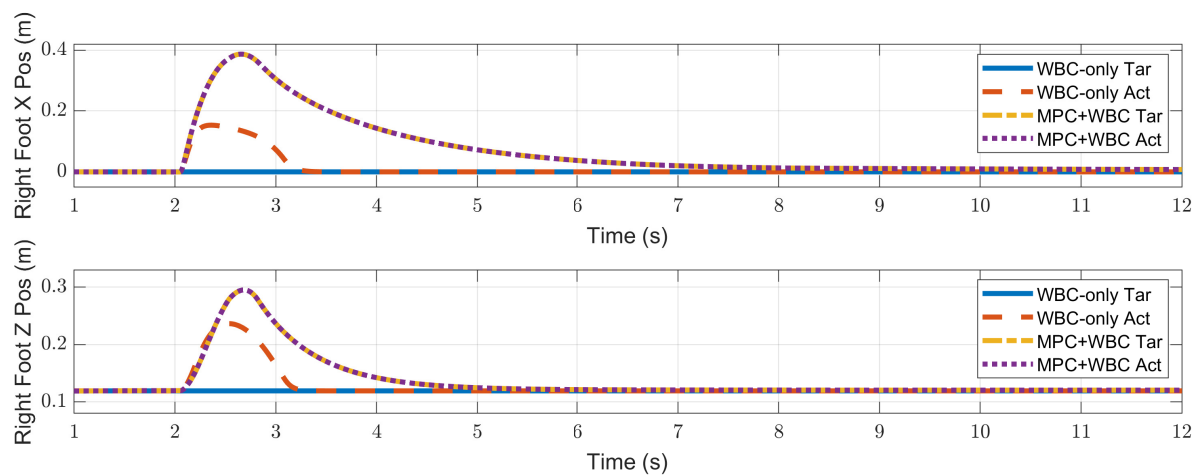


Figure 6. Swing foot position of the robot under the frontal impact of a 9 kg rigid ball.

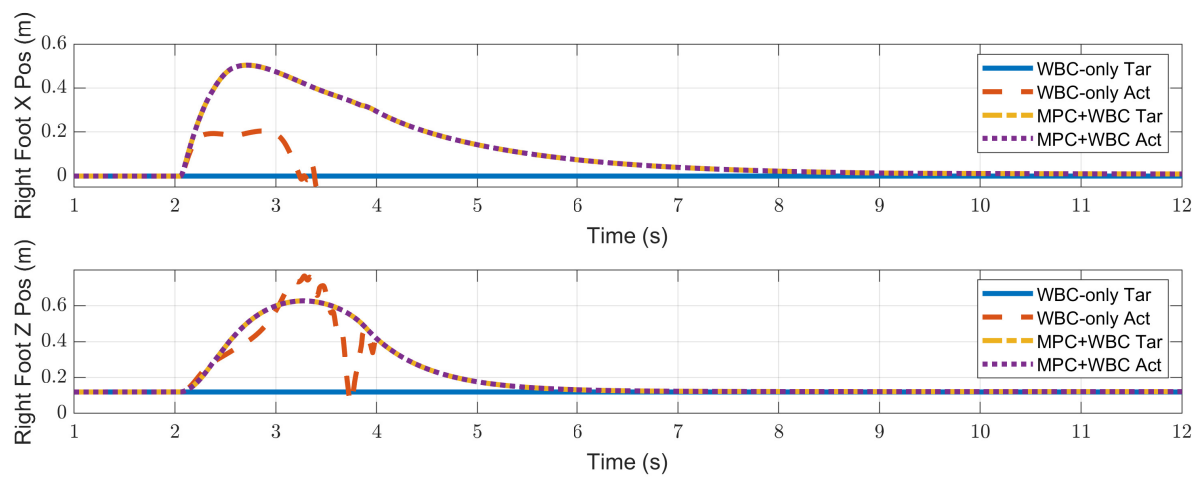


Figure 7. Swing foot position of the robot under the frontal impact of an 11 kg rigid ball.

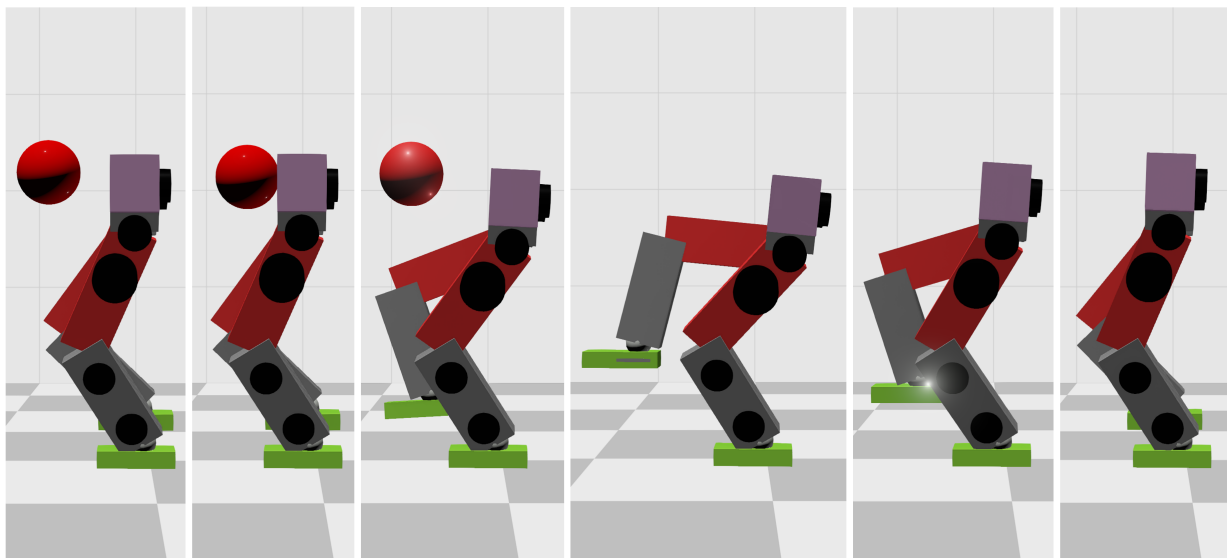


Figure 8. Screenshots of the motions generated by our method in the frontal impact test.

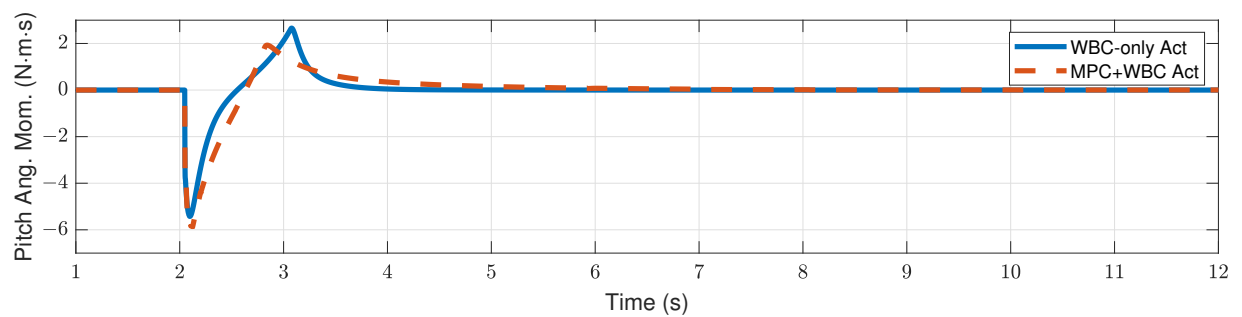


Figure 9. Centroidal pitch angular momentum of the robot under the frontal impact of a 9 kg rigid ball.

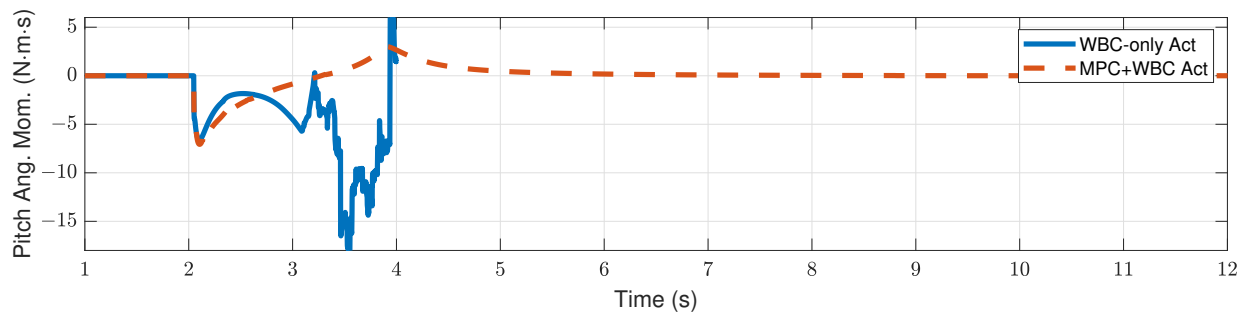


Figure 10. Centroidal pitch angular momentum of the robot under the frontal impact of an 11 kg rigid ball.

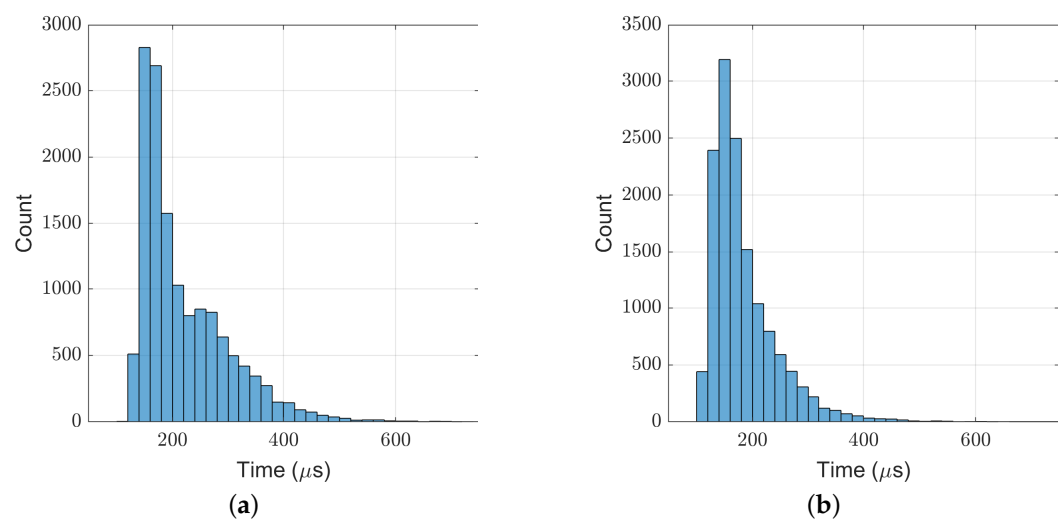


Figure 11. Computation time cost of TP-MPC in the frontal impact test. (a) 9 kg ball; (b) 11 kg ball.

4.2.2. Side Impact

Compared with frontal impact, the robot is more vulnerable to side impact since the supporting polygon is narrower in the side direction. For side impact, when the ball's mass is 2.7 kg or 3.2 kg, the robot's CoM trajectories during the impact and recovery stages are shown in Figures 12 and 13, respectively. Figures 14 and 15 show the actual trajectories of the swing leg's foot end and the correspondent reference trajectories given to WBC. Figures 16 and 17 show the robot's centroidal roll angular momentum. Similar to the situation of frontal impact in Section 4.2.1, both control schemes can balance the robot under the 2.7 kg ball's impact, with our MPC-WBC scheme achieving better performance. Under the 3.2 kg ball's impact, the WBC-only scheme fails, while our scheme still balances the robot. Screenshots in Figure 18 show that the robot can also side-swing its leg under a side impact. Figure 19 shows the computation time cost of the TP-MPC algorithm when the robot is subjected to a side impact.

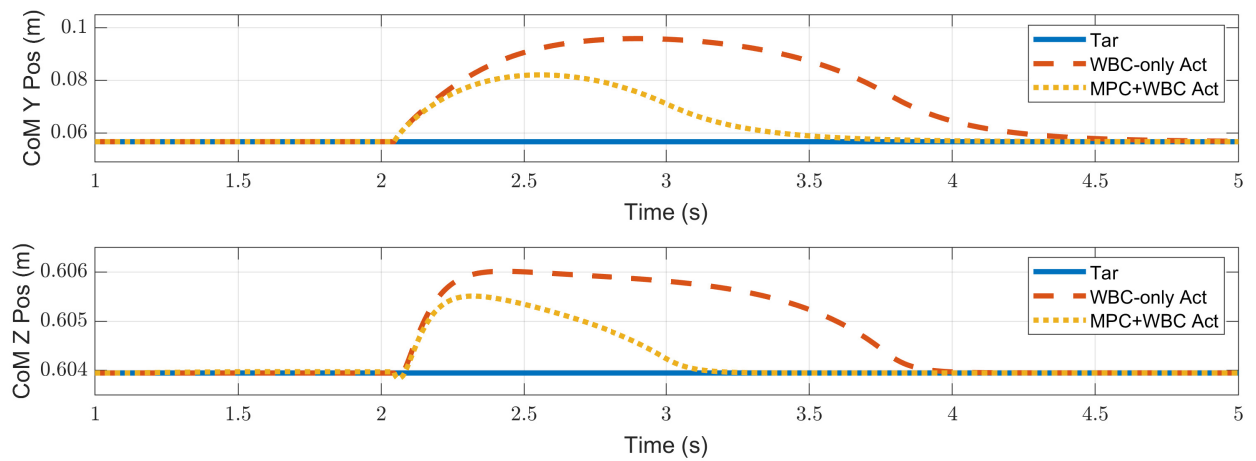


Figure 12. CoM position of the robot under the side impact of a 2.7 kg rigid ball.

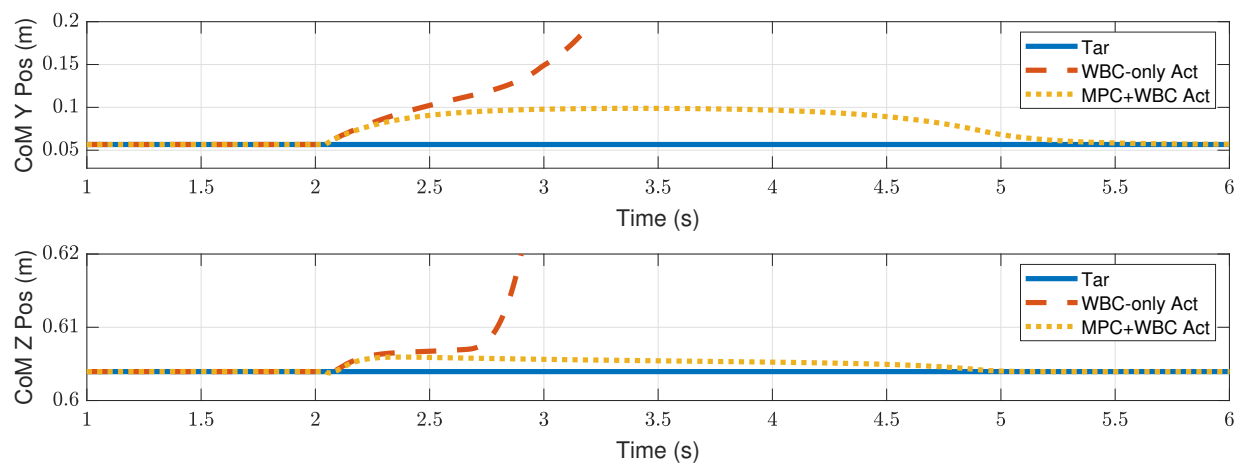


Figure 13. CoM position of the robot under the side impact of a 3.2 kg rigid ball.

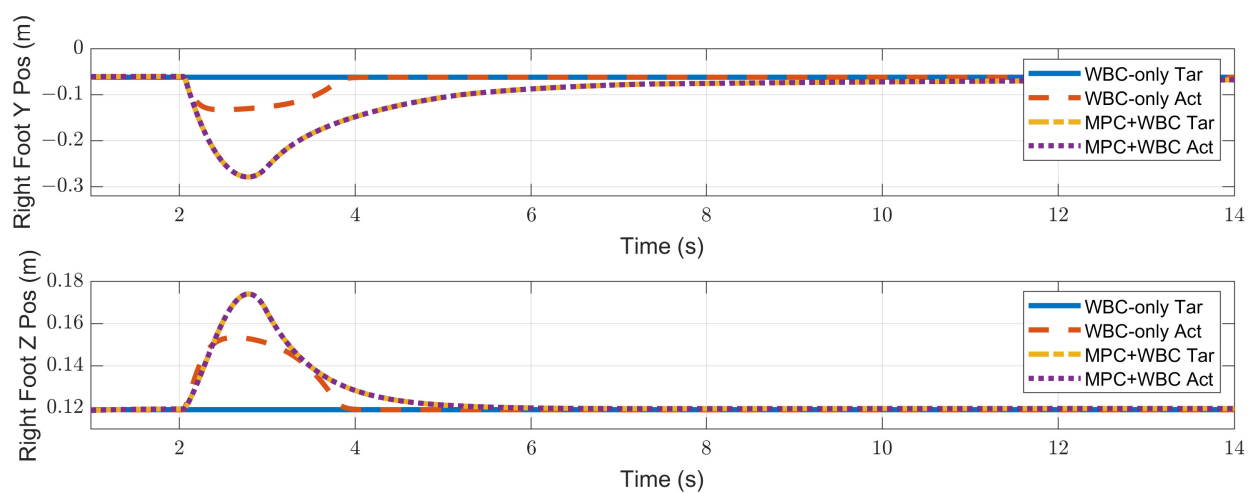


Figure 14. Swing foot position of the robot under the side impact of a 2.7 kg rigid ball.

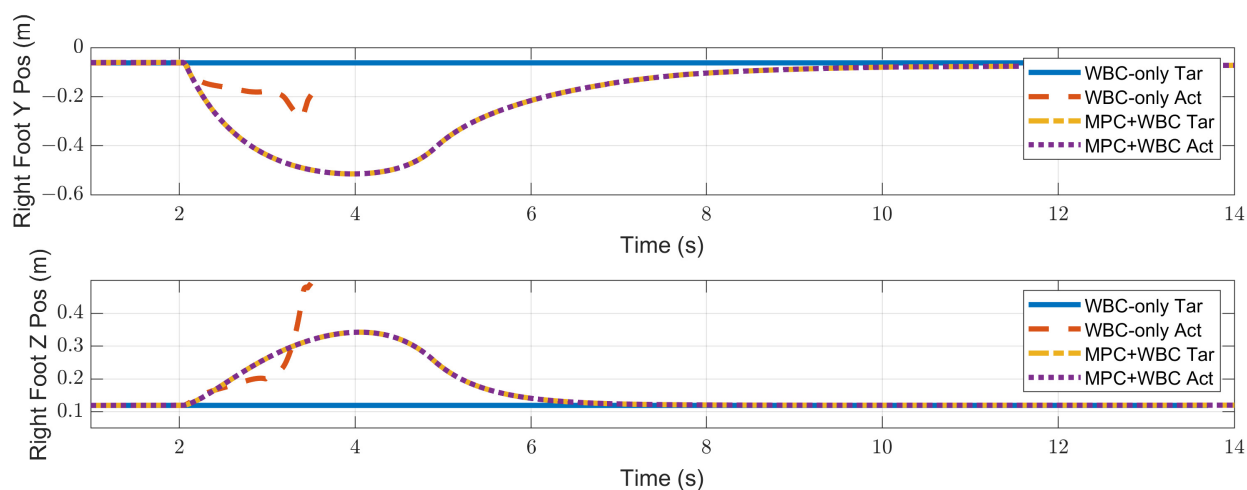


Figure 15. Swing foot position of the robot under the side impact of a 3.2 kg rigid ball.

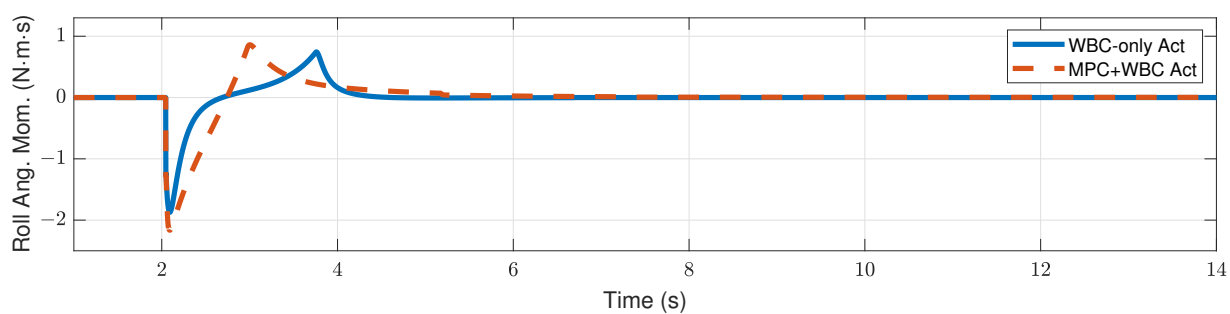


Figure 16. Centroidal roll angular momentum of the robot under the side impact of a 2.7 kg rigid ball.

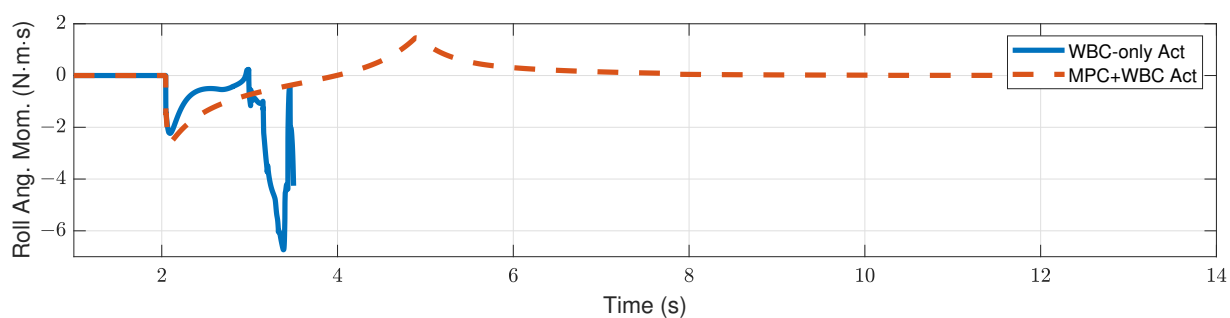


Figure 17. Centroidal roll angular momentum of the robot under the side impact of a 3.2 kg rigid ball.

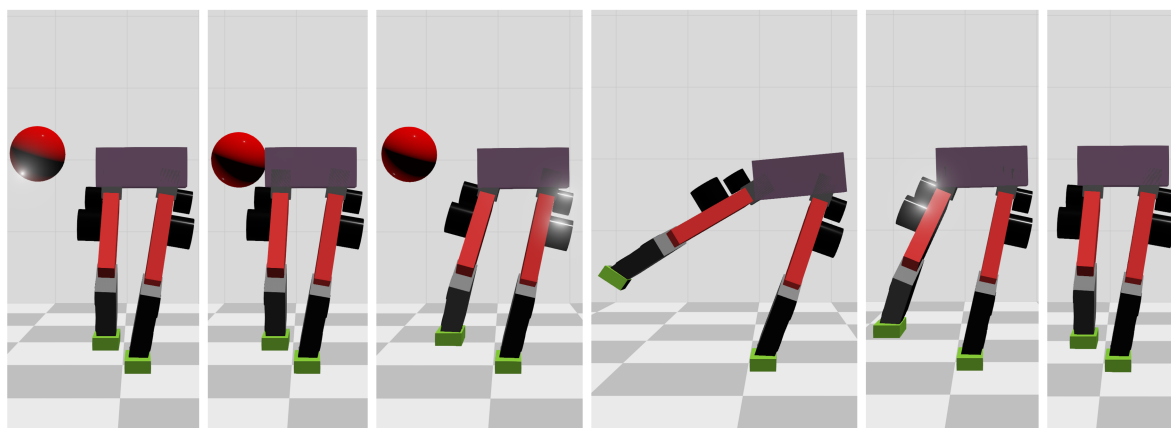


Figure 18. Screenshots of the motions generated by our method in the side impact test.

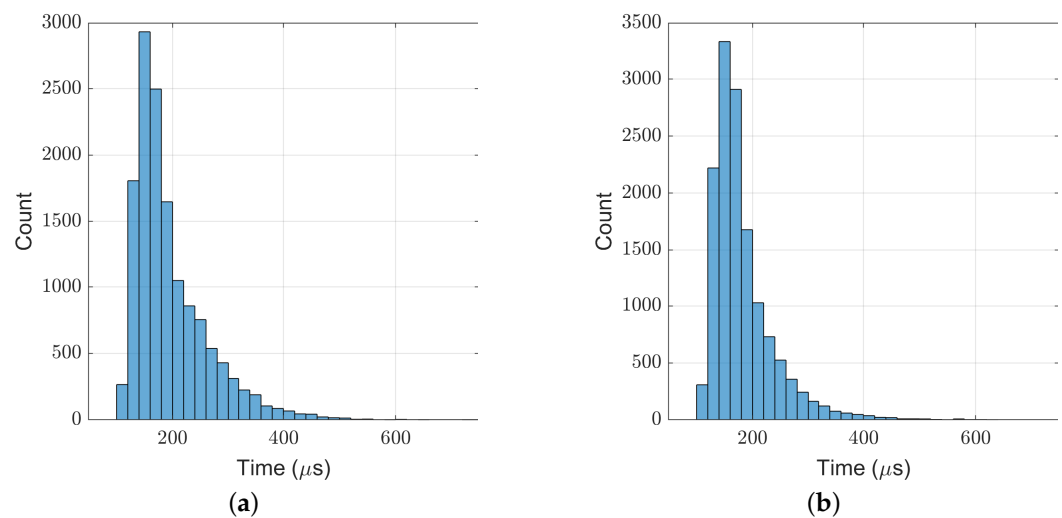


Figure 19. Computation time cost of TP-MPC in the side impact test. (a) 2.7 kg ball; (b) 3.2 kg ball.

4.3. Discussion

In the WBC, the centroidal dynamics task is considered the second-priority task. Due to its existence, the biped robot can mobilize all joints to maintain the position and velocity of the CoM. Therefore, the WBC-only controller has a certain degree of anti-interference ability. However, it may fail to deal with a considerable impact. The centroidal dynamics task is expressed on the acceleration level. When the acceleration required to achieve the second-priority tasks has been satisfied, the third-priority tasks will be executed as much as possible. That is to say, the WBC will execute the task of swing foot to some extent, even if the robot's CoM has not returned to its initial state. In the WBC-only control scheme, the desired trajectory of the swing foot only keeps the initial state unchanged, which is almost unhelpful for maintaining the stability of the CoM. On the contrary, in our control scheme, the TP-MPC provides the swing foot with the desired motion trajectory that can balance the position of the CoM. The task of swing foot is almost consistent with the centroidal dynamics task in WBC. Therefore, the control scheme with TP-MPC shows a better ability to maintain the stability of the center of mass.

In the above experiments, the maximum computation time of our TP-MPC is 961 μs , and the average value is about 200 μs . Through proper optimization in coding, our algorithm can run at the frequency of 1 kHz on the real robot.

5. Conclusions

This paper proposes the three-particle model predictive control scheme to solve the balance control problem of a bipedal robot standing on one foot. We verified our control scheme in simulation. Compared with the WBC-only control scheme, our scheme can resist much larger disturbances and bring the robot back to its initial state much faster. Although the TP-MPC discards the complicated full dynamics of the robot, it can still catch the main effects of the swing leg motions on the robot's CoM. Furthermore, since our MPC employs only linear dynamics and constraints, it requires little computational cost, implying real-time execution at the same frequencies as the WBC on a real robot.

Our controller also has some deficiencies. The three-particle model may be oversimplified in some cases, for example, where the robot's legs or torso have large angular velocities. In the future, we plan to improve our simplified model, for example, including the rigid body's rotational inertia or adding more particles to the model. We also plan to verify our method on an actual bipedal robot.

Author Contributions: Conceptualization, Y.Y. and J.S.; methodology, Y.Y., W.C., S.H. and Q.L.; software, Y.Y., J.S., W.C. and Q.L.; validation, Y.Y.; formal analysis, Y.Y. and J.S.; investigation, Y.Y. and J.S.; resources, M.Z.; data curation, Y.Y., J.S. and Y.G.; writing—original draft preparation, Y.Y., J.S., S.H. and Y.G.; writing—review and editing, M.Z., Y.Y., J.S., S.H., Y.G., W.C., X.C. and X.L.; visualization, Y.Y. and Y.G.; supervision, M.Z. and X.L.; project administration, M.Z.; funding acquisition, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Innovation 2030-Key Project under grant 2021ZD0201402.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Luo, R.C.; Huang, C.W.; Hung, W.C. Bipedal robot push recovery control mimicking human reaction. In Proceedings of the 2016 IEEE 14th International Workshop on Advanced Motion Control (AMC), Auckland, New Zealand, 22–24 April 2016; pp. 334–339. [\[CrossRef\]](#)
2. Ficht, G.; Behnke, S. Direct Centroidal Control for Balanced Humanoid Locomotion. In Proceedings of the 25th Climbing and Walking Robots Conference, Ponta Delgada, Portugal, 12–14 September 2022; Springer: Cham, Switzerland, 2022; pp. 242–255.
3. Raza, F.; Zhu, W.; Hayashibe, M. Balance stability augmentation for wheel-legged biped robot through arm acceleration control. *IEEE Access* **2021**, *9*, 54022–54031. [\[CrossRef\]](#)
4. Sentis, L.; Khatib, O. A whole-body control framework for humanoids operating in human environments. In Proceedings of the Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, Orlando, FL, USA, 15–19 May 2006; pp. 2641–2648.
5. Ju, X.; Wang, J.; Han, G.; Zhao, M. Mixed Control for Whole-Body Compliance of a Humanoid Robot. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 8331–8337. [\[CrossRef\]](#)
6. Kim, D.; Di Carlo, J.; Katz, B.; Bledt, G.; Kim, S. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv* **2019**, arXiv:1909.06586.
7. Koenemann, J.; Del Prete, A.; Tassa, Y.; Todorov, E.; Stasse, O.; Bennewitz, M.; Mansard, N. Whole-body model-predictive control applied to the HRP-2 humanoid. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3346–3351.
8. Erez, T.; Lowrey, K.; Tassa, Y.; Kumar, V.; Kolev, S.; Todorov, E. An integrated system for real-time model predictive control of humanoid robots. In Proceedings of the 2013 IEEE 13th IEEE-RAS International conference on humanoid robots (Humanoids), Atlanta, GA, USA, 15–17 October 2013; pp. 292–299.
9. Farshidian, F.; Jelavic, E.; Satapathy, A.; Gifftthaler, M.; Buchli, J. Real-time motion planning of legged robots: A model predictive control approach. In Proceedings of the 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), Birmingham, UK, 15–17 November 2017; pp. 577–584.
10. Neunert, M.; Stäuble, M.; Gifftthaler, M.; Bellicoso, C.D.; Carius, J.; Gehring, C.; Hutter, M.; Buchli, J. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1458–1465. [\[CrossRef\]](#)
11. Ding, Y.; Pandala, A.; Park, H.W. Real-time model predictive control for versatile dynamic motions in quadrupedal robots. In Proceedings of the IEEE 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8484–8490.
12. Nguyen, Q.; Powell, M.J.; Katz, B.; Carlo, J.D.; Kim, S. Optimized Jumping on the MIT Cheetah 3 Robot. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 7448–7454. [\[CrossRef\]](#)
13. Kajita, S.; Hirukawa, H.; Harada, K.; Yokoi, K. *Introduction to Humanoid Robotics*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 101.
14. Gibson, G.; Dosunmu-Ogunbi, O.; Gong, Y.; Grizzle, J. Terrain-Aware Foot Placement for Bipedal Locomotion Combining Model Predictive Control, Virtual Constraints, and the ALIP. *arXiv* **2021**, arXiv:2109.14862.
15. Bledt, G.; Kim, S. Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 6316–6323.

16. Vukobratovic, M.; Frank, A.A.; Juricic, D. On the stability of biped locomotion. *IEEE Trans. Biomed. Eng.* **1970**, BME-17, 25–36. [[CrossRef](#)] [[PubMed](#)]
17. Kajita, S.; Tani, K. Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, IEEE Computer Society, Sacramento, CA, USA, 9–11 April 1991; pp. 1405–1406.
18. Pratt, J.; Carff, J.; Drakunov, S.; Goswami, A. Capture point: A step toward humanoid push recovery. In Proceedings of the IEEE 2006 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, 4–6 December 2006; pp. 200–207.
19. Lee, S.H.; Goswami, A. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Auton. Robot.* **2012**, 33, 399–414. [[CrossRef](#)]
20. Xie, Y.; Wang, J.; Dong, H.; Ren, X.; Huang, L.; Zhao, M. Dynamic Balancing of Humanoid Robot with Proprioceptive Actuation: Systematic Design of Algorithm, Software, and Hardware. *Micromachines* **2022**, 13, 1458. [[CrossRef](#)] [[PubMed](#)]
21. Kim, D.; Jorgensen, S.J.; Lee, J.; Ahn, J.; Luo, J.; Sentis, L. Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control. *Int. J. Robot. Res.* **2020**, 39, 936–956. [[CrossRef](#)]
22. Wieber, P.B. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In Proceedings of the 2006 IEEE 6th IEEE-RAS International Conference on Humanoid Robots, Genova, Italy, 4–6 December 2006; pp. 137–142.
23. Li, J.; Nguyen, Q. Force-and-moment-based model predictive control for achieving highly dynamic locomotion on bipedal robots. In Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 14–17 December 2021; pp. 1024–1030.
24. Luo, R.C.; Chen, C.C. Biped Walking Trajectory Generator Based on Three-Mass With Angular Momentum Model Using Model Predictive Control. *IEEE Trans. Ind. Electron.* **2016**, 63, 268–276. [[CrossRef](#)]
25. Boston Dynamics' Humanoid Robot Atlas Shows Off Its Balancing Skills. Available online: <https://laughingsquid.com/boston-dynamics-humanoid-robot-atlas-shows-off-its-balancing-skills/> (accessed on 9 December 2022).
26. Kanoun, O.; Lamiroux, F.; Wieber, P.B.; Kanehiro, F.; Yoshida, E.; Laumond, J.P. Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 2939–2944. [[CrossRef](#)]
27. Wensing, P.M.; Orin, D.E. Improved Computation of the Humanoid Centroidal Dynamics and Application for Whole-Body Control. *Int. J. Humanoid Robot.* **2016**, 13, 1550039:1–1550039:23. [[CrossRef](#)]
28. Bi, Y.; Gao, J.; Lu, Y.; Cao, J.; Zuo, W.; Mu, T. Simulation of Improved Bipedal Running Based on Swing Leg Control and Whole-body Dynamics. In Proceedings of the 2021 6th International Conference on Robotics and Automation Engineering (ICRAE), Guangzhou, China, 19–22 November 2021; pp. 134–140. [[CrossRef](#)]
29. Lu, Y.; Gao, J.; Shi, X.; Tian, D.; Jia, Z. Whole-Body Control Based on Landing Estimation for Fixed-Period Bipedal Walking on Stepping Stones. In Proceedings of the 2020 3rd International Conference on Control and Robots (ICCR), Tokyo, Japan, 26–29 December 2020; pp. 140–149. [[CrossRef](#)]
30. Michel, O. Webots: Professional Mobile Robot Simulation. *J. Adv. Robot. Syst.* **2004**, 1, 39–42.
31. Felis, M.L. RBDL: An efficient rigid-body dynamics library using recursive algorithms. *Auton. Robot.* **2017**, 41, 495–511. [[CrossRef](#)]
32. Ferreau, H.; Kirches, C.; Potschka, A.; Bock, H.; Diehl, M. qpOASES: A parametric active-set algorithm for quadratic programming. *Math. Program. Comput.* **2014**, 6, 327–363. [[CrossRef](#)]