Journal of Intelligent & Robotic Systems (2022) 105: 6
https://doi.org/10.1007/s10846-022-01618-z

REGULAR PAPER

Online Robust Gait Generator of Biped Robots Inspired by Human Anti-disturbance Strategies

Jingchao Li[1] · Zhaohui Yuan[1] · Sheng Dong[1] · Yingxing Li[1] · Jianrui Zhang[1] · Fuli Zhang[1]

Received: 31 December 2020 / Accepted: 10 March 2022 / Published online: 20 April 2022

Abstract
In order to enhance the anti-disturbance ability of biped robots, a novel linear model predictive control framework is proposed in this paper. We integrate the step duration, footstep location, and angular momentum into the objective function while the center of pressure(CoP) is located in the supporting polygon. The contributions of this paper are as follows. First, four anti-disturbance strategies are applied online simultaneously. The anti-disturbance ability is improved compared to the methods that only consider three or less strategies. Second, we use known initial values to avoid the nonlinear constraints caused by adjusting the step duration. The optimization problem is converted to standard quadratic programming(QP), thus reducing the computational complexity. Simulation results show that this approach has stronger anti-disturbance ability than the previous linear anti-disturbance methods while ensuring low time cost.

Keywords Biped walking · Anti-disturbance · Model predictive control · Linear reaction wheel pendulum · Online gait generation · Humanoid robots

1 Introduction

Biped robots have been designed to achieve all kinds of human locomotion abilities. At present, biped robots have been able to achieve stable walking in the laboratory environment. However, in the specific working environment, various external forces will disturb robots inevitably, which will cause sudden changes in momentum. Therefore, it is of great importance to have an online gait generator with strong anti-disturbance ability. However, the dynamics of biped robots have the characteristics of high dimension, nonlinear, and hybrid. Under the existing computing resources, it is difficult to design an online anti-disturbance controller for their whole-body dynamics. Therefore, the hierarchical control framework [12, 13] has been widely studied, which uses simplified models to generate top-level trajectories. Kajita et al. [19] proposed to generate gait online by using the cart-table model combined with model predictive control. This method could carry out CoP real-time feedback control. Wieber et al. [35] further improved the CoP preview control scheme. The center of mass(CoM) jerk was taken as the optimization input to resist disturbance to a certain extent. Englsberger et al. [11] and Takenaka et al. [33] presented a gait generation method based on the divergent component of motion (DCM), which is the divergence in the linear inverted pendulum model (LIPM). It was controlled to ensure the balance of the walking process. Kim et al. [22] further developed the method by using three mass linear inverted pendulum model [30]. However, when encountering strong disturbance, they can only adjusted CoP or DCM, which cannot maintain a good anti-disturbance performance.

When humans are disturbed by external forces during working, there are usually four strategies, respectively are moving CoP by ankle torque, adjusting the footstep location, changing the step duration, and rotating the upper limbs to change the angular momentum of the CoM. For changing the footstep location, Herdt et al. [16, 17] and Diedam et al. [8] wrote the footstep location into the objective function, which allowed the robot to adapt to the footstep location under disturbance. Morisawa et al. [26] got adaptive CoP fluctuation and the modification of footstep location by using the position and speed deviation of center of gravity and method of numerical optimization. Castano et al. [5] used the feedback of the centroid state to modify the footstep location online. For utilizing the

✉ Jingchao Li
jingchaoli@mail.nwpu.edu.cn

[1] School of Automation, Northwestern Polytechnical University, Xi'an 710072, China

angular momentum, Goswami et al. [14] proposed to make the angular momentum change rate zero, and extended the stability criterion for the centroid to include angular momentum. Hofmann et al. [18] enhanced control of the CoM by using angular momentum. At the same time, angular momentum also received a lot of attention in whole-body motion control [7, 28]. For changing step duration, because the position of CoM is a nonlinear function of step duration, Bohorquez et al. [3] used the method of setting bounds to ensure the iterations meet nonlinear constraints. Maximo et al. [25] used mixed-integer quadratic programming to solve the nonlinear problem. Wang et al. [34] limited DCM to stable boundaries, and then a variable step duration gait under fixed foot locations constraints was implemented. In short, the above approaches all improved the anti-disturbance ability during walking to a certain extent, but only two strategies have been considered. Moreover, they all used nonlinear method to adjust the step duration, which increased the computation complexity.

In recent years, more and more researchers studied the comprehensive use of multiple strategies. Aftab et al. [1] proposed a nonlinear MPC framework to generate ankle, hip and stepping strategies under adjustable step duration. Ding et al. [9] and Kryczka et al. [24] considered both footstep location and the step duration, and used nonlinear optimization (NLP) to solve the problem. However, NLP often leads to too long time to solve, and may not guarantee to find the optimal solution, while requiring higher hardware resources. Stephens et al. [32] proposed a push recovery model predictive control method and implemented whole-body step recovery control for robots with force-controlled joints. In [31], the objective was reformulated to include the instantaneous capture point(ICP) [29], but the angular momentum and step duration were not considered. Kojio et al. [23] gave a method to maintain balance by adjusting step duration and angular momentum with limited support area. Dong et al. [10] proposed a flexible model predictive control approach and Chappell et al. [6] combined two asynchronous real-time optimizations. They could realize the online adjustment of footstep location and step duration [15, 21], but they did not consider the effect of angular momentum.

In this paper, a novel linear MPC control framework is proposed. By using the linear reaction wheel pendulum (LRWP) model [27], a linear simplification of the reaction wheel pendulum model [2], a speed tracking online gait generator is designed. In the prediction time domain, the rotation angle, footstep location, step duration, and CoM trajectory are obtained, the variation of CoP and the centroid rotation angle acceleration are used as inputs to minimize the objective function. The footstep location and the step duration can deviate from the target value set artificially according to the reference speed input. The rotation angle can be adapted according to the situation. Moreover, we design the unique linear CoP constraints, structural constraints, and driving constraints under this MPC control framework. The known optimized initial values are utilized to achieve adjustable step duration under linear constraints. In the end, the MPC computation is designed as QP to ensure real-time online computation with lower computing resources.

The rest of this paper is organized as follows. In Section 2, the preliminaries are briefly reviewed, the LRWP model and system optimization model are introduced. Then, in Section 3, we give the objective function and constraints, and demonstrate how this optimization problem can be transformed into QP to be solved quickly online. After then, Section 4 makes the systematic comparative evaluations with existing linear methods and robustness analysis by simulation. The conclusion and future works are summarized in Section 5.

2 Preliminaries

2.1 LRWP Dynamics

In the hierarchical control framework of biped robots, the LIPM [20] has been widely used because of its linear characteristic. Compared to the LIPM, the LRWP model adds angular momentum and maintains its original linear property. The added angular momentum can represent the influence of the upper limb rotation, and make it more reflective of the dynamic characteristics of the robot walking process.

The dynamic equation of the CoM of the robot can be expressed as:

$$m(\ddot{\boldsymbol{c}} + \boldsymbol{g}) = \sum_i \boldsymbol{f}_i \tag{1}$$

$$\dot{\boldsymbol{L}} = \sum_i (\boldsymbol{p}_i - \boldsymbol{c}) \times \boldsymbol{f}_i \tag{2}$$

where m is the overall mass, c denotes the position of CoM, L is the angular momentum. p and f are the contact point and contact force. The subscript i is the index of the contact points. g is the acceleration of gravity.

As shown in Fig. 1, we take the z-direction perpendicular to the ground, the x-direction is the forward direction, and the y-direction is the side swing direction. Take the height of the contact point between the robot and the ground as zero($p_i^z = 0$) and keep the height of the CoM constant($\ddot{c}^z = 0$), the expression of Eqs. 1 and 2 in x-and y-directions is:

$$\boldsymbol{c}^{x,y} - \frac{c^z}{g}\ddot{\boldsymbol{c}}^{x,y} + \frac{1}{mg}\boldsymbol{S}\dot{\boldsymbol{L}}^{x,y} = \frac{\sum_i f_i^z \boldsymbol{p}_i^{x,y}}{\sum_i f_i^z} \tag{3}$$

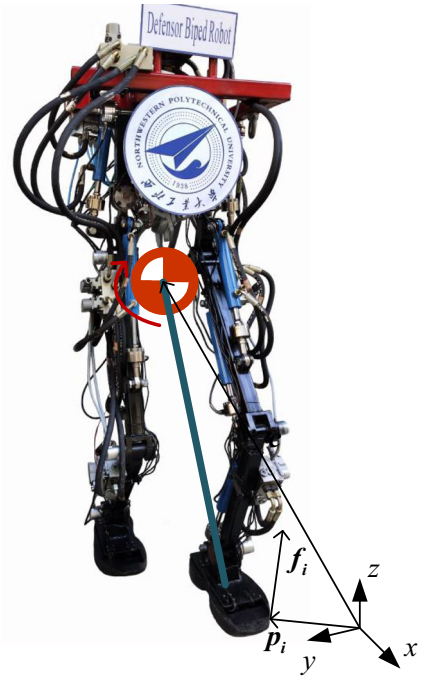

Fig. 1 Defensor hydraulic biped robot

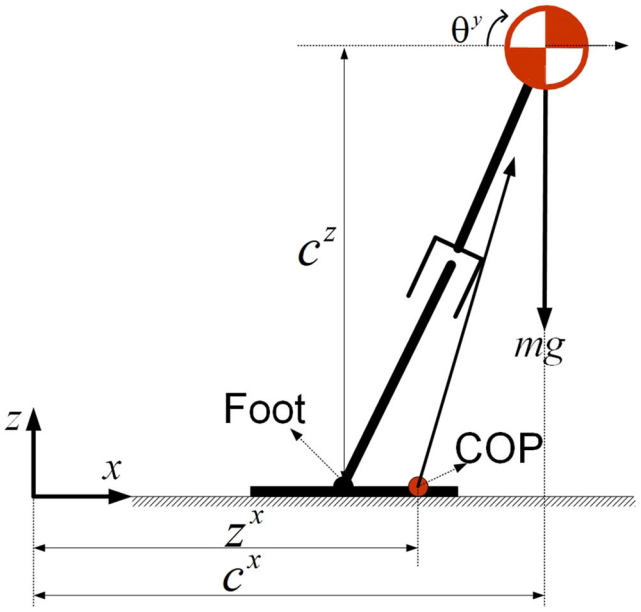

Fig. 2 LRWP model(x-direction)

where $\boldsymbol{S} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, the superscript represents the relative coordinate axes.

Let $\omega = \sqrt{g/c^z}$, Eq. 3 is reformulated as:

$$\ddot{\boldsymbol{c}}^{x,y} = \omega^2(\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}) + \frac{\omega^2}{mg}\boldsymbol{S}\dot{L}^{x,y} \tag{4}$$

where z is the position of CoP.

$$\boldsymbol{z}^{x,y} \triangleq \frac{\sum\limits_i f_i^z \boldsymbol{p}_i^{x,y}}{\sum\limits_i f_i^z} \tag{5}$$

Assuming the inertia of the robot relative to the center of mass as a constant($\dot{L}^x = I^x\ddot{\theta}^x$, $\dot{L}^y = I^y\ddot{\theta}^y$), we can get the decoupling expression of Eq. 4 in the x-and y-directions.

$$\ddot{c}^x = \omega^2(c^x - z^x) - \sigma^y\ddot{\theta}^y \tag{6}$$

$$\ddot{c}^y = \omega^2(c^y - z^y) + \sigma^x\ddot{\theta}^x \tag{7}$$

where I denotes the inertia, θ is the rotation angle relative to the centroid. $\sigma^y = \frac{I^y}{mc^z}$, $\sigma^x = \frac{I^x}{mc^z}$. Equations 6 and 7 are the dynamics of LRWP model as shown in Fig. 2.

2.2 System Optimization Model

In this section, we iteratively compute the LRWP model in the prediction time domain to obtain the system optimization model for optimization calculation. Since the LRWP model is fully decoupled in both x-and y-directions, only the angular momentum term has a sign difference (the inertial coordinate system is the right-handed coordinate system), and the rest of the dynamics are the same, thus we take the x-direction for deducing.

Discretize Eq. 6 with interval T,

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k^x \tag{8}$$

where the subscript k repersents the time step,

$$\boldsymbol{A} = \begin{bmatrix} 1 + \frac{1}{2}T^2\omega^2 & T & -\frac{1}{2}T^2\omega^2 & 0 & 0 \\ T\omega^2 & 1 & T\omega^2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} -\frac{1}{2}T^2\omega^2 & -\frac{1}{2}T^2\sigma^{y2} \\ -T/\omega^2 & -T \\ 1 & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix},$$

$$\boldsymbol{x} = \begin{bmatrix} c^x \\ \dot{c}^x \\ z^x \\ \theta^y \\ \dot{\theta}^y \end{bmatrix}, \boldsymbol{u}^x = \begin{bmatrix} \Delta z^x \\ \ddot{\theta}^y \end{bmatrix}.$$

Take the length of the prediction time domain as N and the duration as NT, let $\boldsymbol{X} = \left(\boldsymbol{x}_k^T, \boldsymbol{x}_{k+1}^T, \ldots, \boldsymbol{x}_{k+N-1}^T\right)^T$, $\boldsymbol{U}^x = \left[\boldsymbol{u}_k^T, \boldsymbol{u}_{k+1}^T, \ldots, \boldsymbol{u}_{k+N-1}^T\right]^T$, Eq. 8 is iteratively calculated,

$$\boldsymbol{X} = \overline{\boldsymbol{A}}\boldsymbol{x}_k + \overline{\boldsymbol{B}}\boldsymbol{U}^x \tag{9}$$

where,

$$\overline{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^{N-1} \\ \boldsymbol{A}^N \end{bmatrix}, \overline{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} & 0 & \cdots & \cdots & 0 \\ \boldsymbol{AB} & \boldsymbol{B} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ \boldsymbol{A}^{N-2}\boldsymbol{B} & \boldsymbol{A}^{N-3}\boldsymbol{B} & & \boldsymbol{B} & \boldsymbol{0} \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \cdots & \boldsymbol{AB} & \boldsymbol{B} \end{bmatrix}$$

Equation 9 is the system optimization model in x-direction. It should point that Eq. 9 takes a slightly different form in the y-direction which should be calculated based on Eq. 7.

In the following sections, we collectively express the same physical quantity in the prediction time domain. $\boldsymbol{C}^x = \left(c_k^x, c_{k+1}^x, \ldots, c_{k+N-1}^x\right)^T$, $\boldsymbol{\Theta}^x = \left(\theta_k^x, \theta_{k+1}^x, \ldots, \theta_{k+N-1}^x\right)^T$. y-direction is the same.

3 Problem Formulation

3.1 Objective Function

This paper selects the objective function shown in Eq. 10,

$$J = \frac{a_1}{2}\left\|\boldsymbol{C}^{\text{target}} - \boldsymbol{C}\right\|^2 + \frac{a_2}{2}\left\|\dot{\boldsymbol{C}}^{\text{target}} - \dot{\boldsymbol{C}}\right\|^2 + \frac{a_3}{2}\|\boldsymbol{U}\|^2 + \frac{a_4}{2}\left\|\boldsymbol{x}_f^{\text{target}} - \boldsymbol{x}_f\right\|^2 + \frac{a_5}{2}\|\Theta\|^2 + \frac{a_6}{2}\left\|\boldsymbol{T}_{\text{step}}^{\text{target}} - \boldsymbol{T}_{\text{step}}\right\|^2 + \frac{a_4}{2}\left\|\boldsymbol{y}_f^{\text{target}} - \boldsymbol{y}_f\right\|^2 \tag{10}$$

where,

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}^x \\ \boldsymbol{C}^y \end{bmatrix}, \quad \dot{\boldsymbol{C}} = \begin{bmatrix} \dot{\boldsymbol{C}}^x \\ \dot{\boldsymbol{C}}^y \end{bmatrix}, \quad \Theta = \begin{bmatrix} \Theta^y \\ \Theta^x \end{bmatrix}$$

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}^x \\ \boldsymbol{U}^y \end{bmatrix}, \boldsymbol{T}_{\text{step}} = \begin{bmatrix} T_{\text{step}}^x \\ T_{\text{step}}^y \end{bmatrix}, \boldsymbol{x}_f \in \boldsymbol{R}^M, \boldsymbol{y}_f \in \boldsymbol{R}^M$$

$\boldsymbol{x}_f$ and $\boldsymbol{y}_f$ are the positions of the next M footstep locations in the x-and y-directions, respectively. The superscript target represents the target value.

This objective function makes the robot follow the target CoM velocity and position, the target footstep positions and the step duration as far as possible, and the inputs and rotation angle are the smallest. The target is obtained according to the reference speed as follows.

When M=3, the targets are selected as,

$$\boldsymbol{C}^{x,\text{target}} = \begin{bmatrix} \frac{1}{2}\left(x_{s_0} + x_{f_0}\right) + v_{x\text{ref}}T \\ \frac{1}{2}\left(x_{s_0} + x_{f_0}\right) + v_{x\text{ref}}2T \\ \vdots \\ \frac{1}{2}\left(x_{s_0} + x_{f_0}\right) + v_{x\text{ref}}NT \end{bmatrix},$$

$$\boldsymbol{C}^{y,\text{target}} = \begin{bmatrix} \frac{1}{2}\left(y_{s_0} + y_{f_0}\right) + v_{y\text{ref}}T \\ \frac{1}{2}\left(y_{s_0} + y_{f_0}\right) + v_{y\text{ref}}2T \\ \vdots \\ \frac{1}{2}\left(y_{s_0} + y_{f_0}\right) + v_{y\text{ref}}NT \end{bmatrix} \tag{11}$$

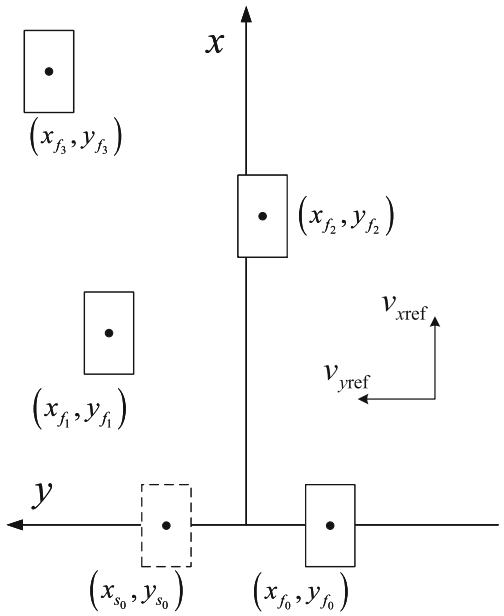

Fig. 3 Predicted footsteps and reference speed

$$\dot{\boldsymbol{C}}^{x,\text{target}} = v_{x\text{ref}}, \dot{\boldsymbol{C}}^{y,\text{target}} = v_{y\text{ref}} \tag{12}$$

$$T_{\text{step}}^{x,\text{target}} = t_{\text{step}}, T_{\text{step}}^{y,\text{target}} = t_{\text{step}} \tag{13}$$

$$\boldsymbol{x}_f^{\text{target}} = \begin{bmatrix} x_{f_0} + t_{\text{step}}v_{x\text{ref}} \\ x_{f_1} + t_{\text{step}}v_{x\text{ref}} \\ x_{f_2} + t_{\text{step}}v_{x\text{ref}} \end{bmatrix},$$

$$\boldsymbol{y}_f^{\text{target}} = \begin{bmatrix} y_{f_0} + y_{\text{step}}\,\text{sign}\left(y_{s_0} - y_{f_0}\right) + t_{\text{step}}\,v_{y\text{ref}} \\ y_{f_1} - y_{\text{step}}\,\text{sign}\left(y_{s_0} - y_{f_0}\right) + t_{\text{step}}\,v_{y\text{ref}} \\ y_{f_2} + y_{\text{step}}\,\text{sign}\left(y_{s_0} - y_{f_0}\right) + t_{\text{step}}\,v_{y\text{ref}} \end{bmatrix} \tag{14}$$

As shown in Fig. 3, $v_{x\text{ref}}$ and $v_{y\text{ref}}$ are the speed reference inputs in x-and y-directions, respectively. t_{step} is the constant reference step duration. y_{step} is the width between the two feet. $\left(x_{s_0}, y_{s_0}\right)$ is the initial position of the swinging foot. $\left(x_{f_0}, y_{f_0}\right)$ is the initial position of the supporting foot. $\left(x_{f_i}, y_{f_i}\right)$ is the footstep location of the i-th step.

In the prediction time domain, we select

$$\boldsymbol{\Phi} = \left[\boldsymbol{U}^{x\,T}, \boldsymbol{x}_f^T, T_{\text{step}}^x, \boldsymbol{U}^{y\,T}, \boldsymbol{y}_f^T, T_{\text{step}}^y\right]^T \in R^{2(2N+M+1)}.$$

The objective function J can be written in the quadratic form of $\boldsymbol{\Phi}$ [4].

$$J = \frac{1}{2}\boldsymbol{\Phi}^T\boldsymbol{H}\boldsymbol{\Phi} + \boldsymbol{f}^T\boldsymbol{\Phi} + J_0 \tag{15}$$

where $\boldsymbol{H} \in \boldsymbol{S}_+^{2(2N+M+1)}$ is Hessian matrix, $\boldsymbol{f} \in R^{2(2N+M+1)}$ is the coefficient of first order , J_0 is a constant independent of $\boldsymbol{\Phi}$. The derivation process is as follows.

Take the x-direction as an example, let

$$\boldsymbol{\Phi}^x = \left[\boldsymbol{U}^{xT}, \boldsymbol{x}_f^T, T_{\text{step}}^x\right]^T \in R^{(2N+M+1)},$$

$$\begin{aligned} J^x =& \frac{a_1}{2}\left\|\boldsymbol{C}^{x,\text{target}} - \boldsymbol{C}^x\right\|^2 + \frac{a_2}{2}\left\|\dot{\boldsymbol{C}}^{x,\text{target}} - \dot{\boldsymbol{C}}^x\right\|^2 \\ &+ \frac{a_3}{2}\left\|\boldsymbol{U}^x\right\|^2 + \frac{a_4}{2}\left\|\boldsymbol{x}_f^{\text{target}} - \boldsymbol{x}_f\right\|^2 + \frac{a_5}{2}\left\|\boldsymbol{\Theta}^y\right\|^2 \\ &+ \frac{a_6}{2}\left\|T_{\text{step}}^{x\ \text{target}} - T_{\text{step}}^x\right\|^2 \end{aligned} \tag{16}$$

J^x can be converted into the following form [4],

$$J^x = \frac{1}{2}\boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{X} + \frac{1}{2}\left(\boldsymbol{U}^x\right)^T\boldsymbol{R}\boldsymbol{U}^x + \frac{1}{2}\boldsymbol{\Gamma}^T\Omega\boldsymbol{\Gamma} + \hat{J}_0^x \tag{17}$$

where, $\boldsymbol{\Gamma} = \left[\boldsymbol{x}_f^T, T_{\text{step}}^x\right]^T$, $\boldsymbol{Q} \in \boldsymbol{S}_+^N$, $\boldsymbol{R} \in \boldsymbol{S}_+^{2N}$, $\Omega \in \boldsymbol{S}_+^{M+1}$, $\hat{J}_0^x$ is a constant. Substitute Eqs. 9 into 17,

$$\begin{aligned} J^x =& \frac{1}{2}\left(\boldsymbol{U}^x\right)^T\left(\boldsymbol{R} + \overline{\boldsymbol{B}}^T\boldsymbol{Q}\overline{\boldsymbol{B}}\right)\boldsymbol{U}^x + \boldsymbol{x}_k^T\overline{\boldsymbol{A}}^T\boldsymbol{Q}\overline{\boldsymbol{B}}\boldsymbol{U}^x + \\ &\frac{1}{2}\boldsymbol{x}_k^T\overline{\boldsymbol{A}}^T\boldsymbol{Q}\overline{\boldsymbol{A}}\boldsymbol{x}_k + \frac{1}{2}\boldsymbol{\Gamma}^T\Omega\boldsymbol{\Gamma} + \hat{J}_0^x. \end{aligned} \tag{18}$$

Let $\boldsymbol{H}^x = \text{diag}\left(R + \overline{\boldsymbol{B}}^T\boldsymbol{Q}\overline{\boldsymbol{B}}, \Omega\right)$, $\boldsymbol{f}^x = [\boldsymbol{x}_k^T\overline{\boldsymbol{A}}^T\boldsymbol{Q}\overline{\boldsymbol{B}}; 0^{M+1}]$, $J_0^x = \frac{1}{2}\boldsymbol{x}_k^T\overline{\boldsymbol{A}}^T\boldsymbol{Q}\overline{\boldsymbol{A}}\boldsymbol{x}_k + \hat{J}_0^x$,

$$J^x = \frac{1}{2}(\boldsymbol{\Phi}^x)^T\boldsymbol{H}^x\boldsymbol{\Phi}^x + (\boldsymbol{f}^x)^T\boldsymbol{\Phi}^x + J_0^x \tag{19}$$

The derivation of the y-direction can be obtained in the same way. Integrating the x-and y-directions, we can get Eq. 15, which is the objective function in two-dimensional.

3.2 Constraints

In the actual walking process, there are four types of constraints. First, constraints on CoP, which ensure the robot does not fall. Second, constraints based on the actual robot structure. For example, the robot cannot cross legs during walking, and the step length cannot exceed the limit of structural size. Third, the single-step movement distance constraints of CoM. These constraints are added to adjust the step duration. Fourth, because the robot drive capability is limited, there are constraints on the upper limb rotation torque and the minimum step duration.

3.2.1 COP Constraints

During the walking process, the CoP must be located within the supporting polygon, and each step includes a double-support phase and a single-support phase. At the beginning of each optimization, the positions of the left foot (x_L, y_L) and the right foot (x_R, y_R) are known. Therefore, we can use the constraints shown in Fig. 4.

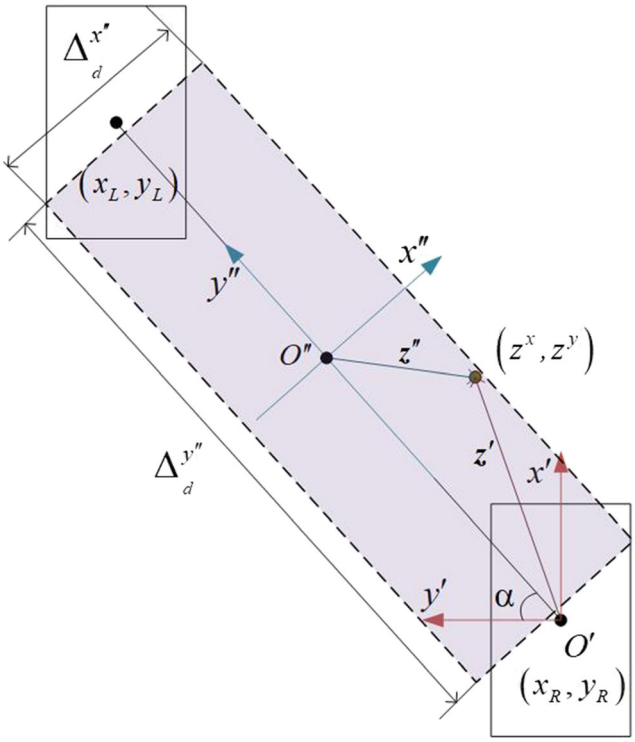

Fig. 4 The first step double support CoP constraints of the prediction time domain

The coordinate of CoP in the coordinate system O'' is $\boldsymbol{z}'' = \left[z^{x''}, z^{y''}\right]^T$.

$$\begin{aligned} z^{x''} &= (z^x - x_R)\cos\alpha - (z^y - y_R)\sin\alpha \\ z^{y''} &= (z^y - y_R)\cos\alpha + (z^x - x_R)\sin\alpha - \frac{l_{\text{pace}}}{2} \end{aligned} \tag{20}$$

where $l_{\text{pace}} = \sqrt{(x_L - x_R)^2 + (y_L - y_R)^2}$, $\alpha = \tan^{-1}\frac{x_L - x_R}{y_L - y_R}$. Equation 20 is reformulated as,

$$\begin{aligned} z^{x''} &= \frac{1}{l_{\text{pace}}}\left[(z^x - x_R)(y_L - y_R) - (z^y - y_R)(x_L - x_R)\right] \\ z^{y''} &= \frac{1}{l_{\text{pace}}}\left[(z^y - y_R)(y_L - y_R) + (z^x - x_R)(x_L - x_R)\right] - \frac{l_{\text{pace}}}{2} \end{aligned} \tag{21}$$

Define the shaded area in Fig. 4 as the stable area. The length and width of the rectangle are $\Delta_d^{y''}$ and $\Delta_d^{x''}$, respectively. The constraints inequalities are,

$$\begin{aligned} -\Delta_d^{x''} \le z^{x''} \le \Delta_d^{x''} \\ -\Delta_d^{y''} \le z^{y''} \le \Delta_d^{y''} \end{aligned} \tag{22}$$

In this paper, we take $\Delta_d^{y''} = 0.35 l_{\text{pace}}$, $\Delta_d^{x''} = 0.3\Delta_d^{y''}$.

In the subsequent continuous walking stage, because the footstep locations are unknown in advance, applying the same constraints as above in the double support phase will lead to nonlinear constraints, which is not conducive to rapid computation. Therefore, the double support phase and

the single support phase are considered together, and the constraints inequalities shown in Eq. 23 are adopted.

$$\begin{aligned} x_{f_i} - \Delta_s^x \le z^x \le x_{f_i} + \Delta_s^x \\ y_{f_i} - \Delta_s^y \le z^y \le y_{f_i} + \Delta_s^y \end{aligned} (i = 1, 2, 3 \cdots, M) \tag{23}$$

where Δ_s^x and Δ_s^y are the length and width of the constraint rectangle, respectively. They are slightly smaller than the length and width of a single foot to ensure the stability.

For the last computation interval in the prediction time domain, Eq. 23 is adjusted so that the CoP is constrained between step M-1 and step M, thus keeping the consistency with the subsequent double support constraints.

$$\begin{aligned} \tfrac{1}{2}(x_{f_{M-1}} + x_{f_M}) - \Delta_s^x \le z^x \le \tfrac{1}{2}(x_{f_{M-1}} + x_{f_M}) + \Delta_s^x \\ \tfrac{1}{2}(y_{f_{M-1}} + y_{f_M}) - \Delta_s^y \le z^y \le \tfrac{1}{2}(y_{f_{M-1}} + y_{f_M}) + \Delta_s^y \end{aligned} \tag{24}$$

When M=2, the COP constraints of the entire prediction time domain is shown in Fig. 5.

3.2.2 Footstep Location Constraints

During walking, the distance between the two footstep locations in the y-direction must be bigger than a certain distance. Otherwise, the legs may interfere with each other. The distance Δ_L^y is determined by the structure of the robot.

$$y_{f_i} - y_{f_{i-1}} \ge \Delta_L^y (i = 1, 2, 3, \cdots, M) \tag{25}$$

3.2.3 Step Length Constraints

The distance between the two footstep locations of the robot in the x-and y-directions must be within certain ranges, namely Δ_F^x and Δ_F^y, which are also determined by the structure of the robot.

$$\begin{aligned} -\Delta_F^x \le x_{f_i} - x_{f_{i-1}} \le \Delta_F^x \\ -\Delta_F^y \le y_{f_i} - y_{f_{i-1}} \le \Delta_F^y \end{aligned} (i = 1, 2, 3, \cdots, M) \tag{26}$$

3.2.4 Rotation Angle Constraints

The rotation around the centroid is completed by upper limbs, but due to kinematic constraints, the rotation angle has upper limits $\Delta_{\theta\max}^x$, $\Delta_{\theta\max}^y$, and lower limits $\Delta_{\theta\min}^x$, $\Delta_{\theta\min}^y$.

$$\begin{aligned} \Delta_{\theta\min}^x \le \theta^x \le \Delta_{\theta\max}^x \\ \Delta_{\theta\min}^y \le \theta^y \le \Delta_{\theta\max}^y \end{aligned} \tag{27}$$

3.2.5 CoM Single-Step Movement Distance Constraints

Take the initial value of the CoM velocity at each optimization as $\dot{c}_0^x$ and $\dot{c}_0^y$, and constrain the movement distance of the CoM for a step duration at this velocity. When the absolute value of the initial velocity is large, the step duration will be reduced to ensure that the constraints will not be violated. Therefore, through the

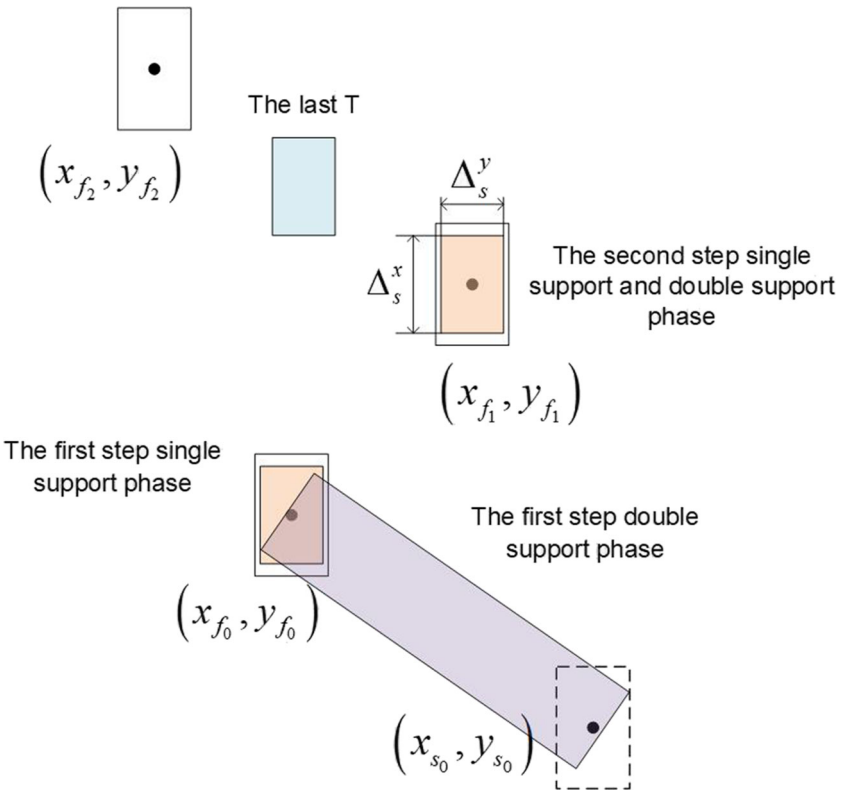

Fig. 5 COP constraints of the prediction time domain

linear constraints, we can adjust the step duration after the robot is disturbed.

$$\begin{aligned} -\Delta_c^x \le \dot{c}_0^x T_{\text{step}}^x \le \Delta_c^x \\ -\Delta_c^y \le \dot{c}_0^y T_{\text{step}}^y \le \Delta_c^y \end{aligned} \tag{28}$$

where Δ_c^x and Δ_c^y are the maximum movement distances in the x-and y-directions, respectively

3.2.6 Torque Constraints

The acceleration of the upper limb rotation is accomplished by a specific electric or hydraulic executor. It is limited by the power of the executor. Therefore, the torque of upper limb rotation has upper limits $\Delta_{\tau\max}^x$, $\Delta_{\tau\max}^y$ and lower limits $\Delta_{\tau\min}^x$, $\Delta_{\tau\min}^y$.

$$\begin{aligned} \Delta_{\tau\min}^x \le \tau^x \le \Delta_{\tau\max}^x \\ \Delta_{\tau\min}^y \le \tau^y \le \Delta_{\tau\max}^y \end{aligned} \tag{29}$$

where the torque is defined as: $\tau^y = I^y \ddot{\theta}^y$, $\tau^x = I^x \ddot{\theta}^x$.

3.2.7 Step Duration Constraints

When the robot is disturbed, it can reduce the next step duration to maintain stability. However, the step duration cannot be infinitely small. Define $\Delta_{T\min}$ as the minimum step duration. These constraints are also determined by the drive capability of the robot.

$$\begin{aligned} T_{\text{step}}^x \le \Delta_{T\min} \\ T_{\text{step}}^y \le \Delta_{T\min} \end{aligned} \tag{30}$$

All of the above constraints are linear constraints on system inputs or system states, which can be reformulated as follows:

$$\boldsymbol{G\Phi} \le \boldsymbol{h} \tag{31}$$

where $\boldsymbol{G}$ and $\boldsymbol{h}$ are constraint coefficients.

3.3 Online Optimization

For the objective function of Eq. 15 and the linear constraints of Eq. 31, neglecting the constant term J_0, the optimization problem can be transformed into a standard QP to realize online fast optimization.

$$\begin{aligned} &\text{minimize } \tfrac{1}{2}\boldsymbol{\Phi}^T \boldsymbol{H\Phi} + \boldsymbol{f}^T \boldsymbol{\Phi} \\ &\text{s.t. } \boldsymbol{G\Phi} \le \boldsymbol{h} \end{aligned} \tag{32}$$

After completing an optimization, we get CoM, CoP, rotation angle trajectories, and M footstep locations of the NT time. The step duration is determined by the smaller of the x-and y-directions.

$$T_{\text{step}}^{x,y} = \min\left(T_{\text{step}}^x, T_{\text{step}}^y\right) \tag{33}$$

During execution, only choose the trajectory of $(M/N)T$ time for the robot to execute. One optimization makes the robot one step forward, and then we perform the next optimization computation. This is repeated to achieve receding horizon control, which is shown in Fig. 6 ($M = 2$).

Before each optimization, the current real state of the robot is estimated and it is used as the initial value of this optimization to achieve online anti-disturbance control. The overall control architecture is shown in Fig. 7. The

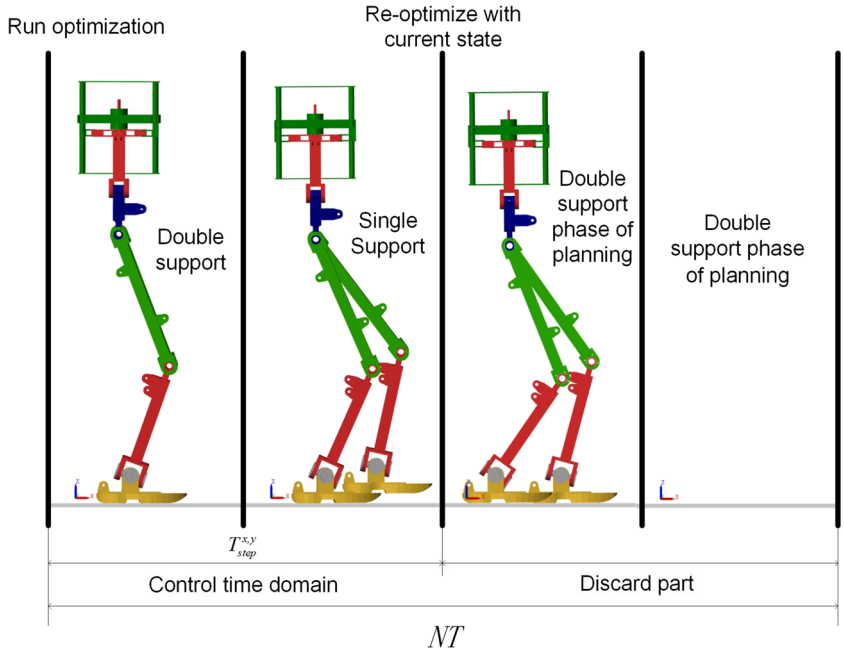

Fig. 6 Two-step prediction time domain

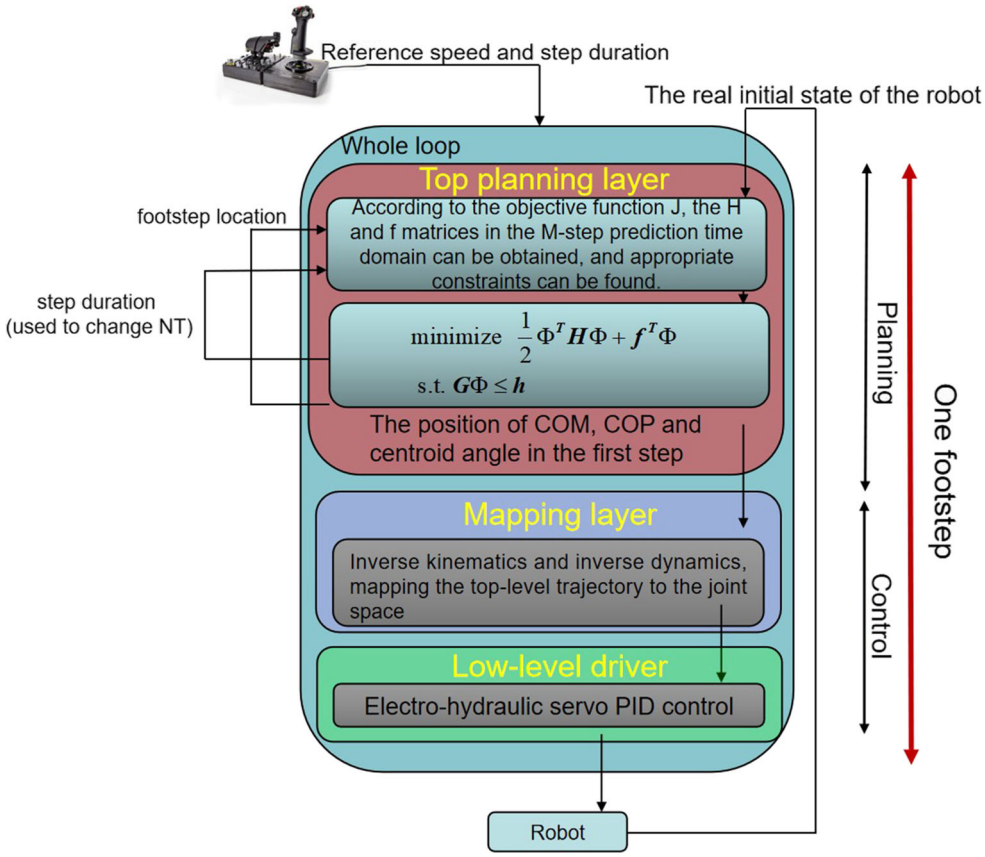

Fig. 7 Overall control architecture

reference input of the robot and the desired step duration are artificially given. Each step is optimized once to optimize the CoM, CoP, and rotation angle trajectories of the future M footsteps. The robot only performs the first step. In the mapping layer, inverse kinematics or inverse dynamics is used to complete the mapping of the top trajectories to the joint space. Finally, the hydraulic servo control technology is used at the bottom to drive the robot joints.

3.4 Computation Efficiency

In this paper, the C++ optimization library QuadProg++ is used to solve the QP. Although the time cost of each control loop varied depending on the initial status, it was less than 45μ s on a single core of an Intel Core i5 @ 3.0 GHz.

Table 1 compares the current state-of-the-art anti-disturbance gait generators, including the anti-disturbance

Table 1 The comparison of the current state-of-the-art anti-disturbance gait generators, including the anti-disturbance strategies they consider, the method used (linear optimization or non-linear optimization), and time cost

Paper	CoP	Step location	Angular momentum	Step duration	Method	Time cost
[25]	√	√		√	Nonlinear	0.1s
[24]	√	√		√	Nonlinear	10ms
[1]	√	√	√	√	Nonlinear	10ms
[9]	√	√		√	Nonlinear	85μ s
[31, 32]	√	√			Linear	30μ s
[10]	√	√		√	Linear	35μ s
This paper	√	√	√	√	Linear	45μ s

strategies they consider, the method used (linear optimization or non-linear optimization), and time cost. The approach proposed in this paper considers four anti-disturbance strategies while ensuring low time cost.

4 Results

In this section, we use simulations to illustrate the push recovery performance of the linear online gait generator which proposed in this paper when subjected to strong external disturbances. Since the time cost of the nonlinear optimization method is significantly greater than that of the linear method, we compare with the two linear gait generators in this section [10, 31].

First, we compare the anti-disturbance ability under the same walking state and disturbance. Then, we conduct a full robustness analysis and comparison with [31] and [10]. The omnidirectional anti-disturbance performance is compared under different walking speed and state. At last, based on the Simscape multibody virtual environment, we verify the feasibility of the algorithm on our self-developed Defensor hydraulic biped robot (Fig. 1).

4.1 Parameters

The model and constraints parameters come from our self-developed Defensor hydraulic biped robot as shown in Fig. 1. The LRWP model parameters are shown in Table 2. The parameters for constraints are shown in Table 3. Table 4 shows the walking and optimization parameters. It should point out that the current robot prototype only has a lower limb structure, and the parameters related to the upper limb are derived from the design. The weights of the optimization problems is tunable parameters to obtain the best performance of each of the three approaches.

4.2 Comparision Under the Same Walking State and Disturbance

In this scenario, the robot starts to walk from the double support phase, which accounts for 10 percent of each footstep. The robot is disturbed when it moves to the second step, which is reflected in the sudden change of the momentum. The disturbance impulse in the x-and y-directions are $I_{x\text{dist}} = 71.4$Ns(Corresponding to the speed change of 0.6m/s) and $I_{y\text{dist}} = -11.9$Ns(Corresponding to the speed change of -0.1m/s), respectively.

The results shown in Figs. 8 and. 9 consider two strategies of adjusting the CoP and the footstep locations [31]. It can be seen from Fig. 8 that after the robot is disturbed in the second step, it adjusts the CoP forward and change the footstep locations of the third step and after steps. However, the robot ultimately fails to return to the reference speed and the CoM diverges in the x-direction.

The results shown in Figs. 10 and 11 consider three strategies of adjusting the CoP, the footstep locations and step duration [10]. Comparing with the method that does not consider the step duration [31], the velocity of the CoM divergence of this method slows down, but it is still cannot restore normal speed tracking.

Figures 12 and 13 show the results obtained by the method proposed in this paper. Four strategies of adjusting CoP, footstep location, step duration, and angular momentum are considered at the same time. After the disturbance at the second step, the CoP is adjusted forward and the footstep locations of the third and fourth steps are adjusted forward and to the right. The step duration of the third step is reduced and the upper body rotates to resist disturbance. Finally, after two steps adjustment, the robot can resume tracking of the reference speed and the rotation of the upper body is also restored after a period of time.

Table 2 LRWP model parameters

Physical quantity	Value
c^z(m)	0.76
m(kg)	119
$I^y(kg \cdot \text{m}^2)$	16
$I^x(kg \cdot \text{m}^2)$	16
g(m/s^2)	9.81

Table 3 Parameters of constraints

Constraints	Value
$\left[\Delta_s^x, \Delta_s^y\right](m)$	[0.127,0.0675]
$\Delta_L^y(m)$	0.25
$\left[\Delta_F^x, \Delta_F^y\right](m)$	[0.3,0.3]
$\left[\Delta_{\theta\,\min}^x, \Delta_{\theta\,\min}^y\right](rad)$	[-0. 175,-0.175]
$\left[\Delta_{\theta\,\max}^x, \Delta_{\theta\,\max}^y\right](rad)$	[0.262,0.262]
$\left[\Delta_c^x, \Delta_c^y\right]$(m)	[0.3,0.3]
$\left[\Delta_{\tau\,\min}^x, \Delta_{\tau\,\min}^y\right](N \cdot m)$	[-160,-160]
$\left[\Delta_{\tau\,\max}^x, \Delta_{\tau\,\max}^y\right](N \cdot m)$	[160,160]
Δ_{Tmin}(s)	0.2

Table 4 Walking and optimization parameters

Physical quantity	Value
T (s)	0.01
$[a_1, a_2, a_3, a_4, a_5, a_6]$	[1e-2,3e-1,0.5,1e-2,0.5,1e-4]
$\left[v_{x\text{ref}}, v_{y\text{ref}}\right]$(m/s)	[0.3,0.15]
t_{step} (s)	0.5

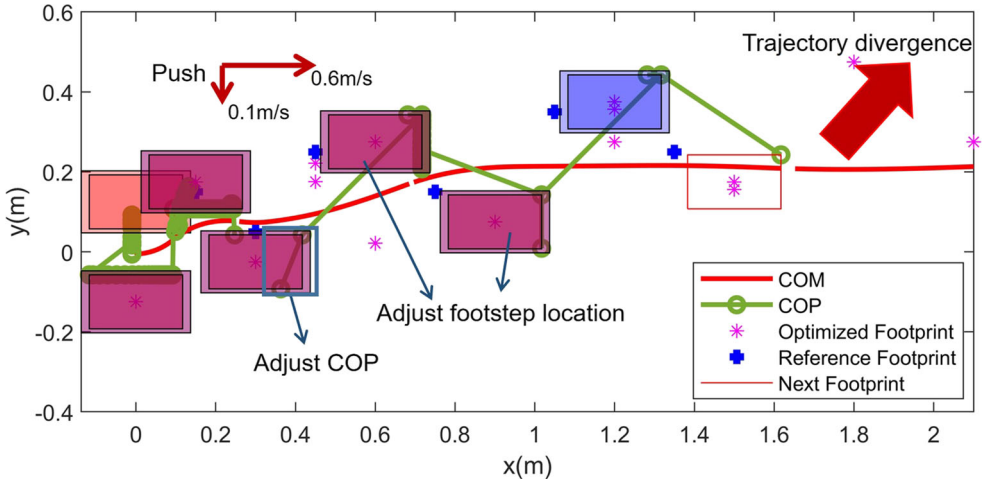

Fig. 8 CoM, CoP trajectories and footstep locations under two anti-disturbance strategies

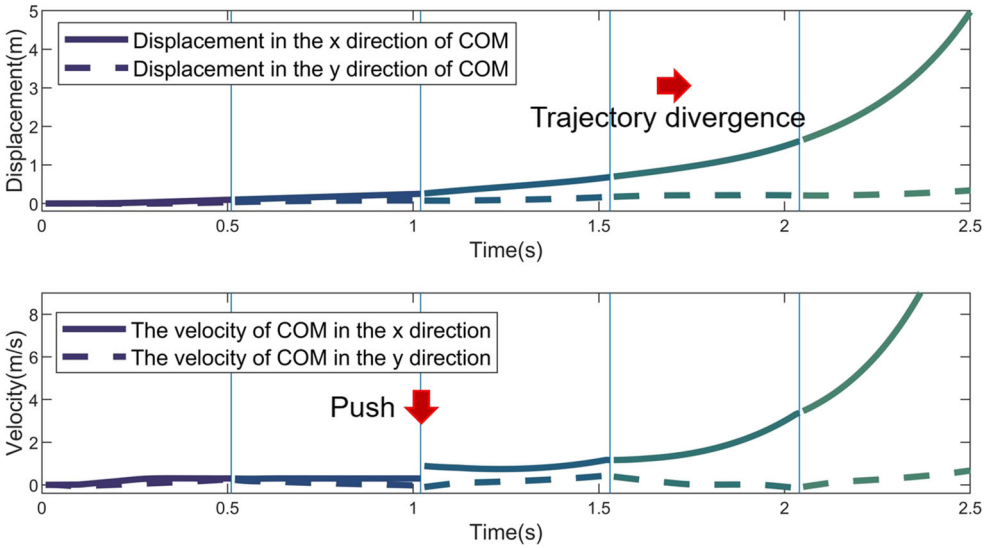

Fig. 9 CoM and CoM velocity under two anti-disturbance strategies

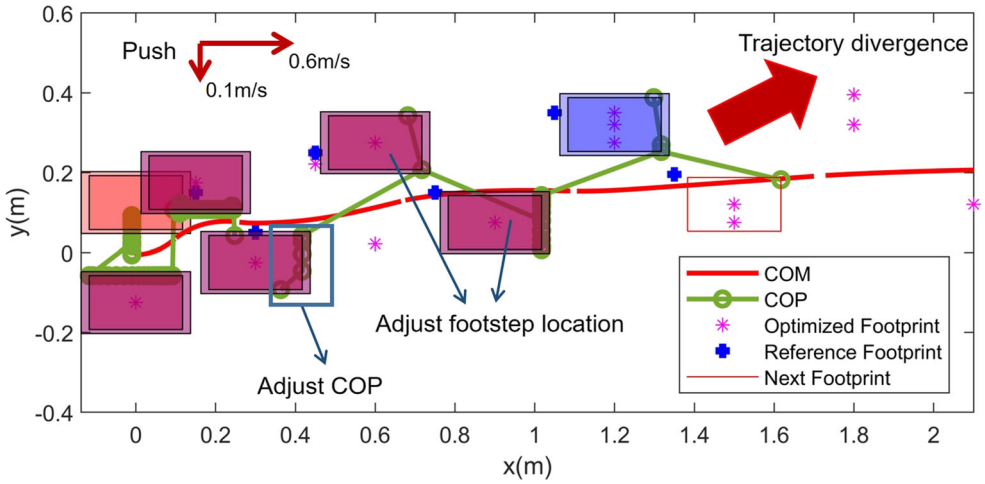

Fig. 10 CoM, CoP trajectories and footstep locations under three anti-disturbance strategies

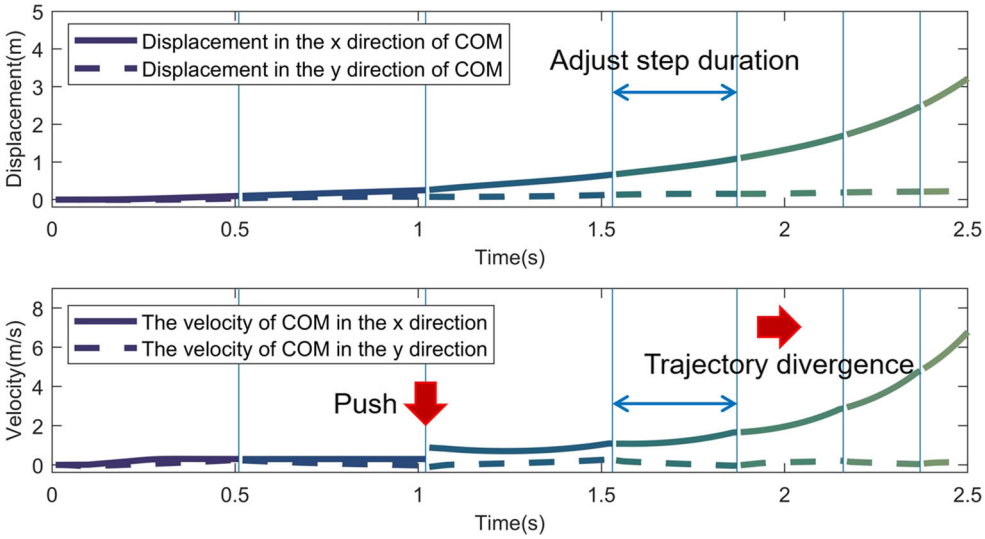

Fig. 11 CoM and CoM velocity under three anti-disturbance strategies

It can be seen that under the walking conditions set by the simulation, the method proposed in this paper shows stronger anti-disturbance performance.

4.3 Robustness Analysis and Comparison

Section 4.2 compares the anti-disturbance performance under specific disturbance and reference gait combination. In actual application, the efficiency of the gait generator greatly depends on the type and speed of the walking state, and the magnitude and direction of the disturbance being applied. In this subsection, we analyze a full set of comparative numerical results under various walking state, magnitude and direction of the disturbance.

The result of the comparison is shown in Fig. 14. The horizontal axis is the reference speed in the x-direction. The vertical axis shows different walking status(Left foot support or right foot support). In each subgraph, the comparison is performed every 30 degrees, and the comparison scale value is 11.9Ns(Corresponding to the speed change of 0.1m/s). Case 1 represents the approach proposed in this paper. Case 2 represents [10] and case 3 [31]. When the disturbance impulse is within the area coverage, the robot will not fall.

It is worth pointing out that although the result in Fig. 14 only changes the reference speed in the x-direction, the same results can be obtained by changing the reference speed in the y-direction.

According to the above simulation, especially the comparison of the omnidirectional anti-disturbance performance in Fig. 14, the approach proposed in this paper is robust to different walking status and disturbances. Moreover, compare to the previous linear anti-disturbance gait generators, the approach proposed in this paper covers the largest stable area in various situations and achieves a comprehensive improvement in anti-disturbance performance.

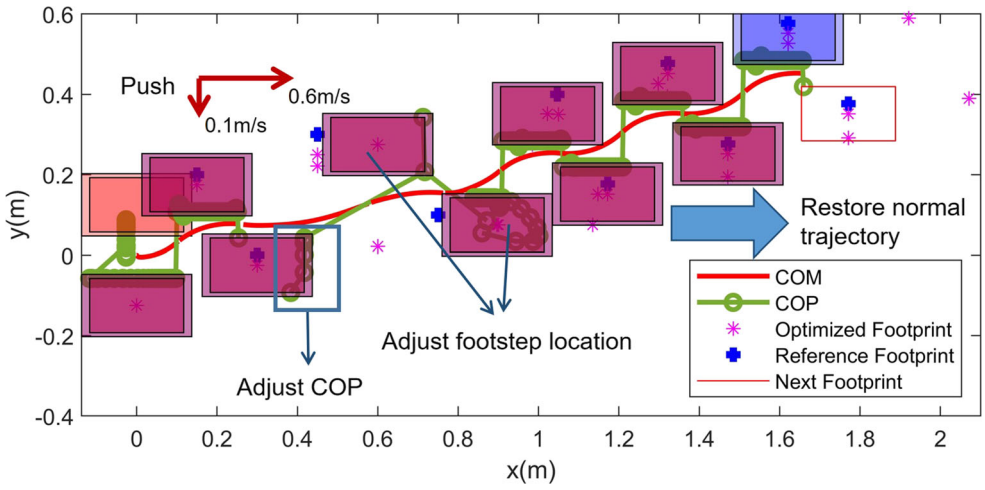

Fig. 12 CoM, CoP trajectories and footstep locations under four anti-disturbance strategies

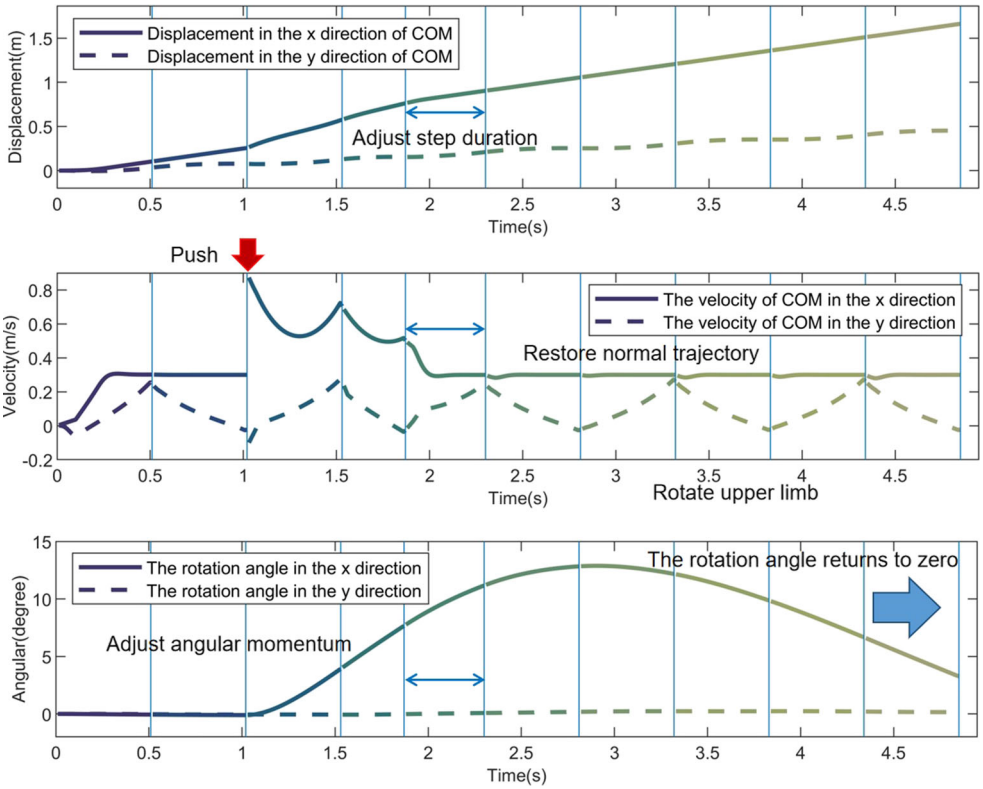

Fig. 13 CoM, CoM velocity and rotation angle under four anti-disturbance strategies

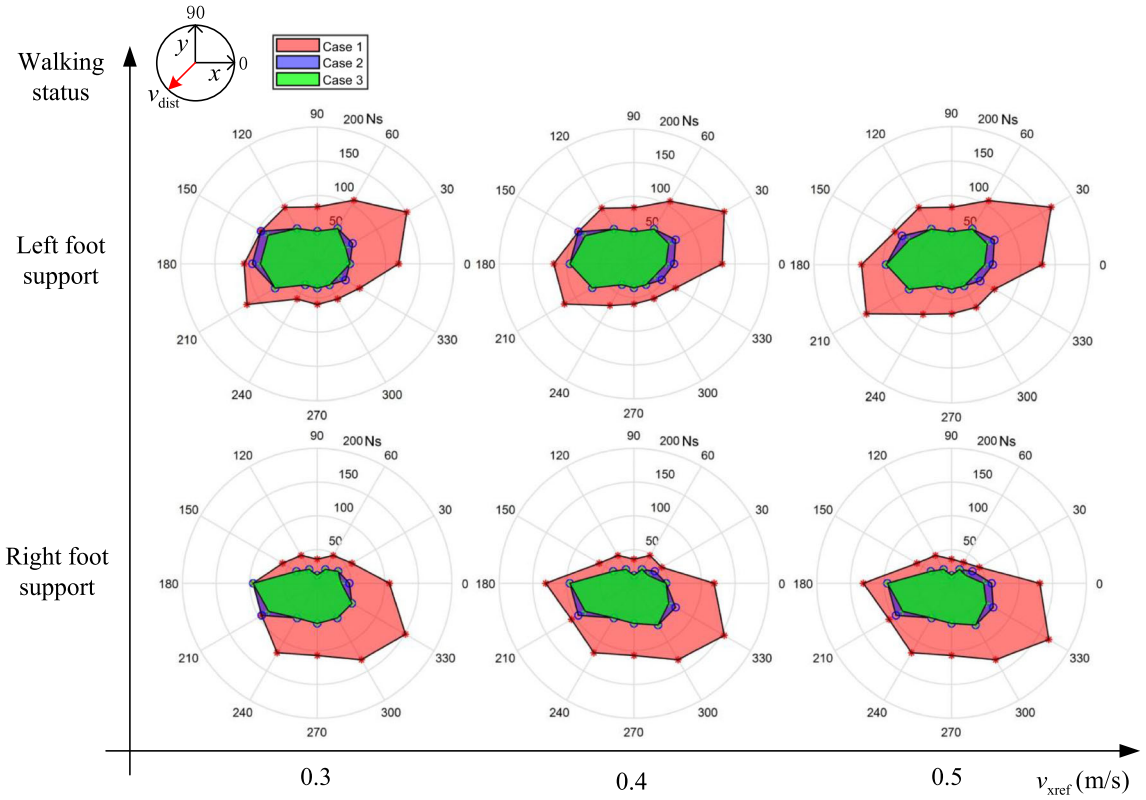

Fig. 14 Comprehensive comparison of anti-disturbance performance

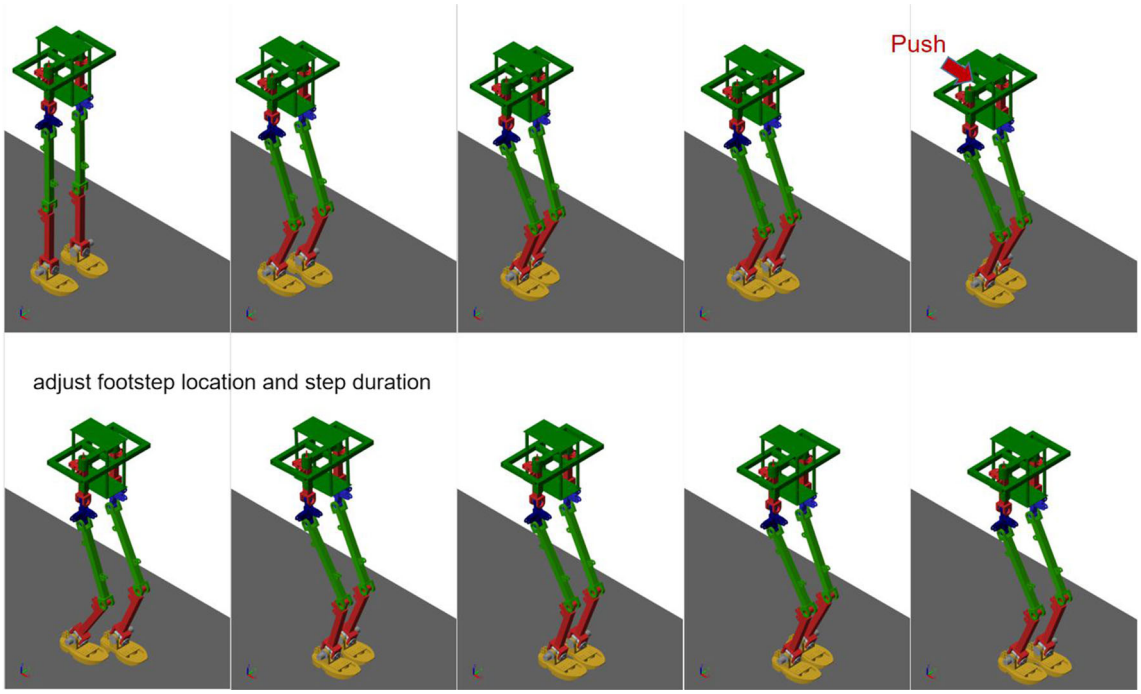

Fig. 15 The visualization of the virtual environment experiment results

4.4 Virtual Environment Simulation

To further demonstrate the effectiveness of the proposed approach, the whole-body dynamic simulation is conducted on the Defensor biped robot by using the Simscape Multibody virtual environment.

The reference walking speeds of the robot is set to $v_{x\text{ref}} = 0.1\text{m/s}$, $v_{y\text{ref}} = 0\text{m/s}$, and the reference step duration is

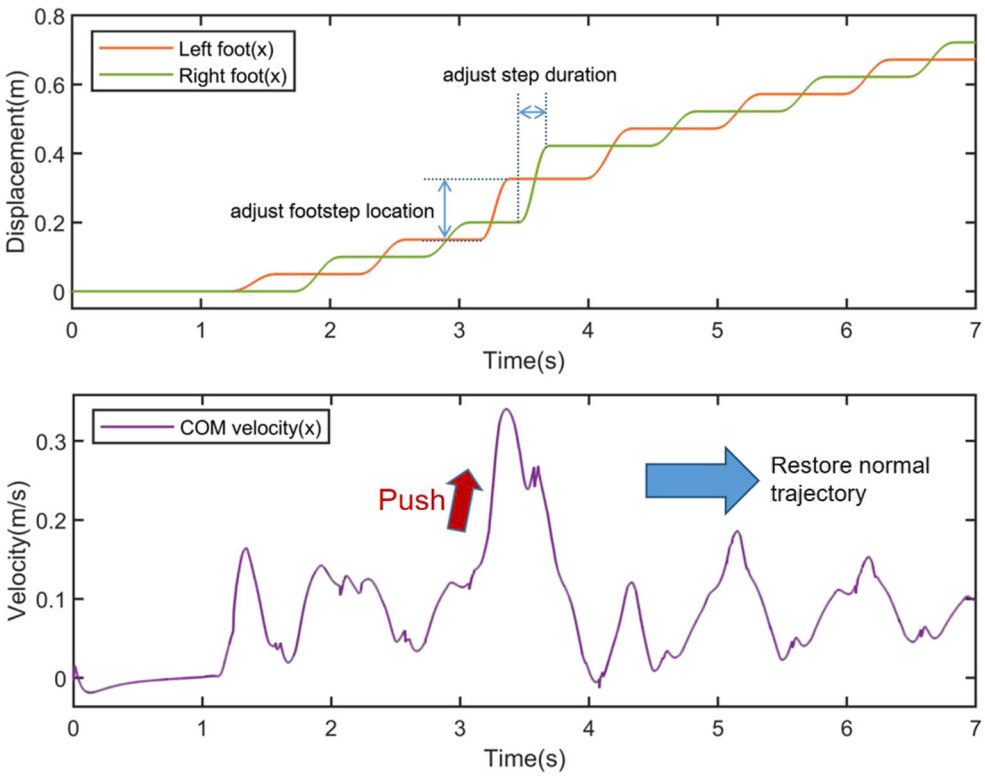

Fig. 16 The trajectories of the feet in the x-direction(upper), the trajectory of the CoM velocity in the x-direction(lower)

$t_{\text{step}} = 0.5$s. When walking to the 3 second, the center of the torso receives a disturbance force in the positive x-axis. The force is 40N and the duration is 0.5s.

Figure 15 shows that the visualization of the experimental result in virtual environment. Figure 16 shows the corresponding physical quantities. The upper picture shows the trajectories of the feet in the x-direction, and the lower picture shows the trajectory of the CoM velocity in the x-direction. When the robot is disturbed, it adjusts its CoP, footstep locations and step durations at the same time in the next two steps. Through the shortened step durations, the footstep locations are adjusted quickly. After two steps of adjustment, the robot restores normal reference speed tracking.

5 Conclusions and Future Works

In general, taking into account four anti-disturbance strategies when human beings are disturbed during walking, the online robust gait generation method based on linear MPC is proposed. This method adjusts CoP, footstep location, step duration, and angular momentum online. Moreover, several flexibly adjustable constraints are given, which can be flexibly adjusted according to the specific robot structure. Compared with previous anti-disturbance algorithms, this method covers the largest anti-disturbance impulse area and achieves a more human-like anti-disturbance posture, which reflects stronger robustness. Last but not least, the problem is transformed into QP, thus ensuring online feedback optimization under fewer computing resources.

In the future, the objective weights which has impact on the performance should be analyzed and auto-tuned, such as [36]. Furthermore, we are to realize the physical verification of the algorithm on the Defensor hydraulic biped robot.

Acknowledgements The authors would like to acknowledge the financial support provided by the Natural Science Basic Research Plan in Shaanxi Province of China (program no. 2018JQ6014), and the Science and Technology program of Gansu Province (program no. 20JR5RA483).

Author Contributions Jingchao Li and Sheng Dong provided the research ideas and the theoretical analysis, wrote the code and the paper; Data curation, Sheng Dong; Formal analysis, Yingxing Li; Funding acquisition, Zhaohui Yuan; Methodology, Jianrui Zhang; Software, Fuli Zhang; Writing original draft, Jingchao Li. All authors have read and agreed to the published version of the manuscript.

Funding The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was financially supported by the Natural Science Basic Research Plan in Shaanxi Province of China (program no. 2018JQ6014), and the Science and Technology program of Gansu Province (program no. 20JR5RA483).

Declarations

Conflict of Interests The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Aftab, Z., Robert, T., Wieber, P.B.: Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme. 2012 12th Ieee-Ras International Conference on Humanoid Robots (Humanoids), 159–164 (2012)
2. Block, D.J., Åström, K.J., Spong, M.W.: The reaction wheel pendulum. Synth. Lect. Control Mechatron. **1**(1), 1–105 (2007)
3. Bohorquez, N., Wieber, P.B.: Adaptive step duration in biped walking: a robust approach to nonlinear constraints. 2017 Ieee-Ras 17th International Conference on Humanoid Robotics (Humanoids), 724–729 (2017)
4. Boyd, S., Boyd, S.P., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)
5. Castano, J.A., Li, Z.B., Zhou, C.X., Tsagarakis, N., Caldwell, D.: Dynamic and reactive walking for humanoid robots based on foot placement control. International Journal of Humanoid Robotics 13(2). https://doi.org/10.1142/S0219843615500413 (2016)
6. Chappell, D., Wang, K., Kormushev, P.: Asynchronous real-time optimization of footstep placement andtiming in bipedal walking robots. arXiv:2007.00385 (2020)
7. Dai, H.K., Valenzuela, A., Tedrake, R.: Whole-body motion planning with centroidal dynamics and full kinematics. 2014 14th Ieee-Ras International Conference on Humanoid Robots (Humanoids), 295–302 (2014)
8. Diedam, H., Dimitrov, D., Wieber, P.B., Mombaur, K., Diehl, M.: Online walking gait generation with adaptive foot positioning through linear model predictive control. 2008 Ieee/Rsj International Conference on Robots and Intelligent Systems, Vols 1-3. Conference Proceedings, 1121–1126 (2008)
9. Ding, J.T., Xiao, X.H.: Tsagarakis, N.: Nonlinear optimization of step duration and step location. 2019 Ieee/Rsj International Conference on Intelligent Robots and Systems (Iros), 2259–2265 (2019)
10. Dong, S., Yuan, Z.H., Yu, X.J., Sadiq, M.T., Zhang, J.R., Zhang, F.L., Wang, C.: Flexible model predictive control based on multivariable online adjustment mechanism for robust gait generation. International Journal of Advanced Robotic Systems 16(6). https://doi.org/10.1177/1729881419887291 (2019)
11. Englsberger, J., Ott, C., Roa, M.A., Albu-Schaffer, A., Hirzinger, G.: Bipedal walking control based on capture point dynamics. 2011 Ieee/Rsj International Conference on Intelligent Robots and Systems, 4420–4427 (2011)
12. Faraji, S.: Towards Robust Bipedal Locomotion: From Simple Models to Full-Body Compliance. Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne (2018)
13. Feng, S.: Online Hierarchical Optimization for Humanoid Control. Ph.D. thesis, Carnegie Mellon University (2016)
14. Goswami, A., Kallem, V.: Rate of change of angular momentum and balance maintenance of biped robots. 2004 Ieee International Conference on Robotics and Automation, Vols 1- 5. Proceedings, 3785–3790. https://doi.org/10.1109/Robot.2004.1308858 (2004)
15. Griffin, R.J., Wiedebach, G., Bertrand, S., Leonessa, A., Pratt, J.: Walking stabilization using step timing and location adjustment on the humanoid robot, atlas. 2017 Ieee/Rsj International Conference on Intelligent Robots and Systems (Iros) pp. 667–673 (2017)

16. Herdt, A., Diedam, H., Wieber, P.B., Dimitrov, D., Mombaur, K., Diehl, M.: Online walking motion generation with automatic footstep placement. Adv. Robot. **24**(5-6), 719–737 (2010). https://doi.org/10.1163/016918610x493552
17. Herdt, A., Perrin, N., Wieber, P.B.: Walking without thinking about it. Ieee/Rsj 2010 International Conference on Intelligent Robots and Systems (IROS), 2010 (2010)
18. Hofmann, A., Popovic, M., Herr, H.: Exploiting angular momentum to enhance bipedal center-of-mass control. Icra: 2009 Ieee International Conference on Robotics and Automation, Vols 1-7 pp. 2483–+ (2009)
19. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of zero-moment point. 2003 Ieee International Conference on Robotics and Automation, Vols 1-3. Proceedings, 1620–1626. https://doi.org/10.1109/robot.2003.1241826 (2003)
20. Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa, H.: The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. Iros 2001: Proceedings of the 2001 Ieee/Rjs International Conference on Intelligent Robots and Systems **1-4**, 239–246 (2001)
21. Khadiv, M., Herzog, A., Moosavian, S.A.A., Righetti, L.: Step timing adjustment: A step toward generating robust gaits. 2016 Ieee-Ras 16th International Conference on Humanoid Robots (Humanoids), 35–42 (2016)
22. Kim, I.S., Han, Y.J., Hong, Y.D.: Stability control for dynamic walking of bipedal robot with real-time capture point trajectory optimization. J. Intell. Robot. Syst. **96**(3-4), 345–361 (2019). https://doi.org/10.1007/s10846-018-0965-7
23. Kojio, Y., Omori, Y., Kojima, K., Sugai, F., Inaba, M.: Footstep modification including step time and angular momentum under disturbances on sparse footholds. IEEE J. Robot. Autom. Lett. **PP**(99), 1–1 (2020)
24. Kryczka, P., Kormushev, P., Tsagarakis, N.G., Caldwell, D.G.: Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing. 2015 Ieee/Rsj International Conference on Intelligent Robots and Systems (Iros), 3352–3357 (2015)
25. Maximo, M.R., Ribeiro, C.H., Afonso, R.J.: Mixed-integer programming for automatic walking step duration. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5399–5404. IEEE (2016)
26. Morisawa, M., Harada, K., Kajita, S., Kaneko, K., Sola, J., Yoshida, E., Mansard, N., Yokoi, K., Laumond, J.P.: Reactive stepping to prevent falling for humanoids. In: 2009 9th IEEE-RAS International conference on Humanoid Robots, pp. 528–534. IEEE (2009)
27. Nenchev, D.N., Konno, A., Tsujita, T.: Humanoid robots: Modeling and control Butterworth-Heinemann (2018)
28. Ponton, B., Herzog, A., Del Prete, A., Schaal, S., Righetti, L.: On time optimization of centroidal momentum dynamics. 2018 Ieee International Conference on Robotics and Automation (ICRA), 5776–5782 (2018)
29. Pratt, J., Koolen, T., De Boer, T., Rebula, J., Cotton, S., Carff, J., Johnson, M., Neuhaus, P.: Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid. Int. J. Robot. Res. **31**(10), 1117–1133 (2012)
30. Sato, T., Sakaino, S., Ohnishi, K.: Real-time walking trajectory generation method with three-mass models at constant body height for three-dimensional biped robots. IEEE Trans. Ind. Electron. **58**(2), 376–383 (2011). https://doi.org/10.1109/Tie.2010.2052535
31. Stephens, B.: Push Recovery Control for Force-Controlled Humanoid Robots. Ph.D. thesis, Carnegie Mellon University (2011)
32. Stephens, B.J., Atkeson, C.G.: Push recovery by stepping for humanoid robots with force controlled joints. In: IEEE-RAS International Conference on Humanoid Robots (2010)
33. Takenaka, T., Matsumoto, T., Yoshiike, T.: Real time motion generation and control for biped robot-1(st) report: Walking gait pattern generation. 2009 Ieee-Rsj International Conference on Intelligent Robots and Systems, 1084–1091. https://doi.org/10.1109/Iros.2009.5354662 (2009)
34. Wang, H.T., Zhao, M.G.: A robust biped gait controller using step timing optimization with fixed footprint constraints. 2017 Ieee International Conference on Robotics and Biomimetics (Ieee Robio) **2017**, 1787–1792 (2017)
35. Wieber, P.B.: Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. 2006 6th IEEE-Ras Int. Conf. Humanoid Robot. **1 and 2**, 137–142 (2006). https://doi.org/10.1109/Ichr.2006.321375
36. Yeganegi, M.H., Khadiv, M., Moosavian, S.A.A., Zhu, J.J., Del Prete, A., Righetti, L.: Robust humanoid locomotion using trajectory optimization and sample-efficient learning. In: 2019 IEEE-RAS 19Th International Conference on Humanoid Robots (Humanoids), pp. 170–177. IEEE (2019)

Jingchao Li received the B.S. and M.S. degrees in control science and engineering from Northwestern Polytechnical University (NPU), Xi'an, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree in control science and engineering at NPU and working on the project of the development of hydraulic humanoid robots. His research interests include humanoid robot locomotion control and state estimation for humanoid robots.

Zhaohui Yuan received the B.S., M.S., and Ph.D. degrees in control engineering from Northwestern Polytechnical University, Xi'an, China, in 1984, 1987, and 2005, respectively. He has been with Northwestern Polytechnical University and worked as a Teaching Assistant, a Lecturer, an Associate Professor, and a Professor, since 1987. He has coauthored more than 50 articles in technical journals and conferences. His research interests include high-precision detection instrumentation and system control engineering, hydraulic system control and test, and flow field analysis of the hydraulic systems. He is a Fellow of the Shaanxi Association for Science and Technology.

Sheng Dong received the B.S. degree from Henan Polytechnic University, in 2011, and the M.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2014, in control engineering, where he is currently pursuing the Ph.D. degree. He was a Software Engineer with the Chinese Academy of Space Technology, responsible for the motion control of large antennas. His research interests include intelligent systems and hydraulic humanoid robot control.

Yingxing Li received the B.s. and M.S. degrees in control science and engineering from Northwestern Polytechnical University (NPU), Xi'an, China, in 2019 and 2022, respectively. Since 2022, she is currently pursuing the Ph.D degree in control science and engineering at NPU and working on the project of development for hydraulic humanoid biped robot. Her research interests include hydraulic humanoid biped robot structural design and compliance control for humanoid robots.

Jianrui Zhang received the B.S. and M.S. degrees in mechatronic engineering from Yanshan University, in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree with Northwestern Polytechnical University, Xi'an, China. He was a Lecturer with the College of Mechanical Engineering, Longdong University. He has published one monograph, published more than ten articles, authorized three patents, and three computer software copyrights. He presided more than two research projects. His research interests include mechatronics and robotics.

Fuli Zhang received the B.S. degree in automatic control from Yanching Institute of Technology, Hebei, China, in 2011. And the M.S. degree in testing technology and automation device from Yanshan university. Hebei, China, in 2015. Where he is currently pursuingthe Ph.D. degree in control science and engineering in Northwestern Polytechnical University, Xi'an, China, in 2017. His current research interests include robot mechanism design, robot control and multi-field simulation.