



Research article

A variable-orientation gait planning method for humanoid robots based on nonlinear model predictive control

Tian Mu*, Junyao Gao, Leilei Xie and Mingyue Jin

School of Mechatronics Engineering, Beijing Institute of Technology, Beijing, 100081, China

* **Correspondence:** Email: 3120195102@bit.edu.cn.

Abstract: Based on nonlinear model predictive control (NMPC), this paper proposes a novel gait generation method for robot turning movement. Unlike traditional approaches that treat the robot's heading as a fixed parameter, this method explicitly incorporates the sequence of heading angles into the NMPC framework, which enables the unified optimization of the center of mass (CoM) trajectory, footstep sequence, and heading angle sequence. This design allows the robot to adapt to relatively high speeds and large turning angles. Finally, the proposed method was validated through simulations of the BHR-FCR humanoid robot model in the CoppeliaSim. The results demonstrate that the robot can stably and smoothly execute turning movements at speeds up to 0.75 m/s in response to velocity commands, which confirms the effectiveness of the generated gait sequences.

Keywords: humanoid robot; nonlinear model predictive control; gait planning; turning motion; heading angle

Mathematics Subject Classification: 93C85, 93B45

1. Introduction

Footstep planning is a core feature that sets humanoid robots apart from wheeled robots and other types. This feature enables humanoid robots to exhibit human-like movement patterns and gives them great potential to replace or assist humans in tasks across fields like disaster rescue, industrial production, and social services [1, 2]. However, most existing methods primarily focus on generating stable gaits that resist disturbances [3] and reduce tracking errors, but overlook the capability of turning movement. Therefore, developing a method for turning movements that can uniformly generate and autonomously optimize footstep positions, the CoM trajectory, and the heading angles based on desired speeds remains essential.

Methods enabling robots to achieve turning movement can be roughly divided into two categories: model-based predictive planning methods and deep reinforcement learning-based methods. Due to the

complexity of humanoid robot dynamics, it is difficult to directly plan the CoM trajectory. Therefore, model-based gait planning methods typically first identify an intermediate variable via simplified models [4–7], then use this intermediate variable as the control variable for planning, and thereby derive the CoM trajectory.

Scianca et al. [8] proposed an intrinsically stable MPC framework for generating gaits that enable turning movement. For this framework, the extended linear inverted pendulum (extended LIP) [9] (flywheel-free, no additional angular momentum) serves as the prediction model, with ZMP (zero moment point) [10] velocities adopted as the control input. This method first requires using an omnidirectional motion model to generate the candidate footstep sequence, which means its heading angles and footsteps are optimized in a stepwise manner. Additionally, the linear constraints on the rectangular kinematically admissible region (related to the previous step's footstep and heading angle) are relatively strict. These two factors result in the generated trajectory potentially being unable to track movements at relatively high speed.

Englsberger et al. proposed the concept of the virtual repellent point (VRP) by performing a simplified force analysis of the humanoid robot's CoM motion [11, 12], and further simplified the CoM trajectory of humanoid robots to the spatial variation process of the divergent component of motion (DCM) [13, 14]. Since the motion of the DCM relative to the VRP shows a divergent tendency, the DCM trajectory is generally derived through a reverse recursion approach. This means that when planning turning trajectories, this method requires predefining the footstep sequence and the heading angles, as well as computing the VRP sequence. If the input heading angle is inappropriate, it may pose challenges to the robot's locomotion stability.

Inspired by human locomotion, deep reinforcement learning (DRL) approaches [15–17] and data-driven gait learning methods [18, 19] offer a promising model-free alternative for addressing these challenges by leveraging prior experiences. Rodriguez et al. [20] proposed a single control policy (a single neural network) that takes reference velocities in the x and y directions and the heading angular velocities as inputs. This policy integrates a curriculum learning strategy with a velocity scheduler to gradually increase task difficulty, thus enabling the robot to achieve turning movement. Singh et al. [21] incorporated the upcoming two target steps (including footstep coordinates and heading angles) into the reward function, allowing the robot to gain turning movement capability through tracking reference footsteps. Both methods require the additional input of heading angle-related information during actual control and training.

Considering the limitations of the aforementioned works, a framework for gait generation based on NMPC is proposed in this paper, aiming to achieve the unified optimization of footsteps, CoM trajectory, and heading angles. This framework is verified via simulations in CoppeliaSim, using the BHR-FCR humanoid robot model as shown in Figure 1. The main contributions of this paper are as follows:

1. In the proposed NMPC framework, the heading angle is expressed using the average CoM velocity, which enables its explicit incorporation into the optimization. This allows the algorithm to directly obtain the heading angles, footstep positions, and the CoM trajectory through a single optimization process.
2. Explicit double support phase center of pressure (CoP) constraints are adopted, which expand the feasible region of NMPC optimization. This also enables the method to flexibly adjust prediction

parameters, including the number of predicted steps within one step and the time ratio between single support phases and double support phases during locomotion.

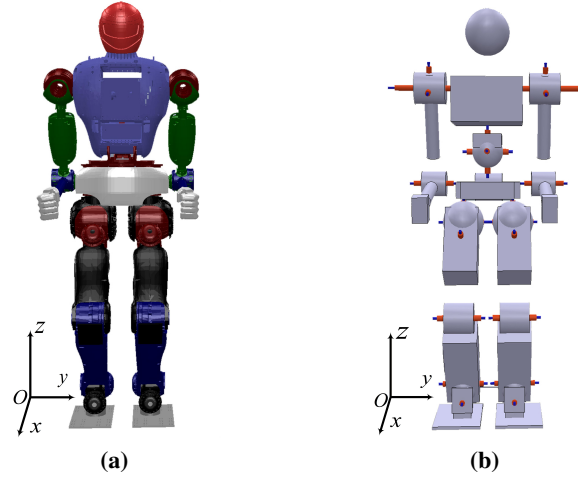


Figure 1. BHR-FCR humanoid robot platform. (a) Simulation model of humanoid robot in CoppeliaSim. (b) Collision box and joint axes of the simulation model.

2. Derivation of the dynamics

This work adopts the widely recognized linear inverted pendulum (LIP) model [22] as the model for describing the robot's dynamics. To simplify the dynamics while retaining key motion characteristics, three core assumptions are made: First, the angular momentum arising from the rotations of all robot components is assumed to be zero; second, the robot's CoM moves within a horizontal plane throughout the locomotion process; third, the normal directions of the contact forces between the feet and the ground are required to be collinear. Based on these assumptions, the robot's motion can be described by three degrees of freedom (DoFs): the horizontal planar coordinates (x, y) of the CoM, and the heading angle θ rotating around the vertical axis z of the world frame.

2.1. Discretization of CoM dynamics

To discretize the robot's CoM dynamics, a constant sampling period T is adopted, and the piecewise constant jerks of CoM in the x and y directions ($\ddot{x}_k, \ddot{y}_k$) are selected as control inputs to ensure the smoothness of the CoM trajectory.

Under this assumption, the state of CoM $\hat{\mathbf{x}}_k = [x_k \ \dot{x}_k \ \ddot{x}_k]^T$ and $\hat{\mathbf{y}}_k = [y_k \ \dot{y}_k \ \ddot{y}_k]^T$ over the prediction horizon can be iteratively derived as

$$\begin{aligned}\hat{\mathbf{x}}_{k+j} &= \mathbf{A}^j \hat{\mathbf{x}}_k + \sum_{i=0}^{j-1} \mathbf{A}^{j-i-1} \mathbf{B} \ddot{x}_{k+i}, \quad j \in [0, N], \\ \hat{\mathbf{y}}_{k+j} &= \mathbf{A}^j \hat{\mathbf{y}}_k + \sum_{i=0}^{j-1} \mathbf{A}^{j-i-1} \mathbf{B} \ddot{y}_{k+i}, \quad j \in [0, N],\end{aligned}\tag{1}$$

$$\mathbf{A} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}. \quad (2)$$

To express the CoM motion over the prediction horizon, the vectors for position, velocity, acceleration, and jerk in the x and y directions are defined as:

$$\begin{aligned} \mathbf{X}_{k+1} &= [x_{k+1}, \dots, x_{k+N}]^T, \mathbf{Y}_{k+1} = [y_{k+1}, \dots, y_{k+N}]^T, \\ \dot{\mathbf{X}}_{k+1} &= [\dot{x}_{k+1}, \dots, \dot{x}_{k+N}]^T, \dot{\mathbf{Y}}_{k+1} = [\dot{y}_{k+1}, \dots, \dot{y}_{k+N}]^T, \\ \ddot{\mathbf{X}}_{k+1} &= [\ddot{x}_{k+1}, \dots, \ddot{x}_{k+N}]^T, \ddot{\mathbf{Y}}_{k+1} = [\ddot{y}_{k+1}, \dots, \ddot{y}_{k+N}]^T, \\ \ddot{\mathbf{X}}_k &= [\ddot{x}_k, \dots, \ddot{x}_{k+N-1}]^T, \ddot{\mathbf{Y}}_k = [\ddot{y}_k, \dots, \ddot{y}_{k+N-1}]^T. \end{aligned} \quad (3)$$

2.2. Center of pressure

As the LIP model adopted in this paper assumes that all mass is concentrated at the CoM, we can derive the CoP position (z_k^x, z_k^y) by using the state vectors $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{y}}_k$:

$$\begin{aligned} z_k^x &= \begin{pmatrix} 1 & 0 & -\frac{h}{g} \end{pmatrix} \hat{\mathbf{x}}_k, \\ z_k^y &= \begin{pmatrix} 1 & 0 & -\frac{h}{g} \end{pmatrix} \hat{\mathbf{y}}_k, \end{aligned} \quad (4)$$

with g denoting the magnitude of gravitational acceleration and h representing the height of the CoM above the ground.

For CoP evolution over the prediction horizon, its position at any prediction step can be iteratively derived using Eq (1) and Eq (4) as

$$\begin{aligned} z_{k+n}^x &= \begin{bmatrix} 1 & 0 & -\frac{h}{g} \end{bmatrix} \hat{\mathbf{x}}_{k+n}, \quad n \in [0, N-1], \\ z_{k+n}^y &= \begin{bmatrix} 1 & 0 & -\frac{h}{g} \end{bmatrix} \hat{\mathbf{y}}_{k+n}, \quad n \in [0, N-1]. \end{aligned} \quad (5)$$

Accordingly, the CoP vectors over the prediction horizon are expressed as

$$\begin{aligned} \mathbf{Z}_{k+1}^x &= [z_{k+1}^x \quad z_{k+2}^x \cdots z_{k+N}^x]^T, \\ \mathbf{Z}_{k+1}^y &= [z_{k+1}^y \quad z_{k+2}^y \cdots z_{k+N}^y]^T. \end{aligned} \quad (6)$$

Here, N represents the prediction horizon length, and each element in $\mathbf{Z}_{k+1}^x$ (or $\mathbf{Z}_{k+1}^y$) corresponds to the CoP's x -coordinate (or y -coordinate) at the n -th future prediction step in the horizon.

2.3. Automatic footstep placement

As reported in [23], tracking reference CoP positions in the objective function can be replaced by directly enforcing constraints on the CoP (which must lie within the support polygon). Meanwhile, the work in [24] provides theoretical analysis of this predictive control scheme, demonstrating that imposing CoP position constraints while minimizing any derivative of the robot's CoM motion enables stable walking motion generation. These insights collectively enable the free generation of feasible footstep positions, which can flexibly adapt to different desired velocities and heading angles.

Meanwhile, rather than tracking a predefined CoP reference position, we set it to be consistent with the footstep position at time step k , which gives

$$\begin{aligned} \mathbf{Z}_{k+1}^{\text{x.ref}} &= \mathbf{V}_{k+1}^{\text{c}} \mathbf{x}_k^{\text{fc}} + \mathbf{V}_{k+1} \mathbf{X}_k^{\text{f}}, \\ \mathbf{Z}_{k+1}^{\text{y.ref}} &= \mathbf{V}_{k+1}^{\text{c}} \mathbf{y}_k^{\text{fc}} + \mathbf{V}_{k+1} \mathbf{Y}_k^{\text{f}}, \end{aligned} \quad (7)$$

where $(x_k^{\text{fc}}, y_k^{\text{fc}})$ denotes the position of the current footstep, and $(\mathbf{X}_k^{\text{f}}, \mathbf{Y}_k^{\text{f}})$ represents the position sequence of the upcoming m footsteps. $\mathbf{V}_{k+1}^{\text{c}} \in \mathbb{R}^N$ and $\mathbf{V}_{k+1} \in \mathbb{R}^{N \times m}$ (N denotes the prediction horizon length, m represents the dimension of footstep-related components, with sampling times corresponding to rows and steps to columns) simply indicate which sampling times t_k fall into which step, and they are given as

$$\mathbf{V}_{k+1}^{\text{c}} := \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{V}_{k+1} := \begin{pmatrix} 0 & 0 & \\ \vdots & \vdots & \\ 0 & 0 & \\ 1 & 0 & \\ \vdots & \vdots & \\ 1 & 0 & \\ 0 & 1 & \\ \vdots & \vdots & \\ 0 & 1 & \ddots \end{pmatrix}. \quad (8)$$

3. Nonlinear model predictive control

In conventional linear walking gait generation methods for humanoid robots, heading angles are often decoupled from the main optimization process [25, 26], computed independently, and then injected into the gait generation framework to linearize the overall problem. Our method differs by formulating heading angles, footstep positions, and center of mass (CoM) trajectory as a unified optimization scheme, which ensures the generated heading angles better align with the robot's walking velocity direction, thereby enhancing the naturalness and adaptability of locomotion. In the following sections, a nonlinear model predictive control (NMPC) method is established to generate stable and adaptive walking gaits.

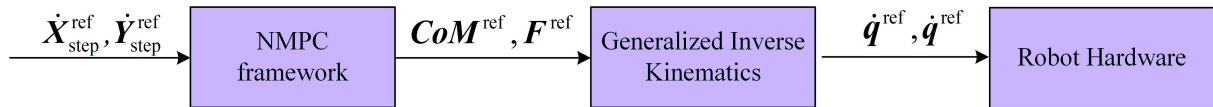


Figure 2. The control scheme. $\dot{\mathbf{X}}_{\text{step}}^{\text{ref}}$ and $\dot{\mathbf{Y}}_{\text{step}}^{\text{ref}}$ act as the input, representing the desired CoM velocity sequences. $\mathbf{CoM}^{\text{ref}}$ and $\mathbf{F}^{\text{ref}}$ are the CoM and the feet trajectories (including position and orientation). $\dot{\mathbf{q}}^{\text{ref}}$ and $\dot{\mathbf{q}}^{\text{ref}}$ denote the robot's joint angle vector and angular velocity vector.

3.1. The controller

A scheme of controller is shown in Figure 2: An open-loop controller is adopted to track the desired CoM velocity sequences given by the user and the reference heading angle sequence derived therefrom. The former are given as sequences in the x and y directions, which are denoted as

$$\dot{\mathbf{X}}_{\text{step}}^{\text{ref}} = [\dot{x}_{\text{step},1}^{\text{ref}}, \dot{x}_{\text{step},2}^{\text{ref}}, \dots, \dot{x}_{\text{step},M}^{\text{ref}}]^T, \quad \dot{\mathbf{Y}}_{\text{step}}^{\text{ref}} = [\dot{y}_{\text{step},1}^{\text{ref}}, \dot{y}_{\text{step},2}^{\text{ref}}, \dots, \dot{y}_{\text{step},M}^{\text{ref}}]^T. \quad (9)$$

Each sequence element corresponds to the average reference velocity for a single step, and M denotes the total number of walking steps planned in the method.

First, the NMPC framework proposed in this paper is used to generate footstep positions, foot orientation angles, the CoM trajectory, and robot heading angles. Based on these outputs, a fifth-order polynomial interpolation method is employed to retrieve the 3D position and orientation trajectories of the feet. Using the same interpolation method, the trajectory of the hip's orientation vectors is generated to ensure the upper body moves consistently with the lower body during locomotion. Meanwhile, to simplify motion planning, we follow a common practice of equating the robot's CoM position to its hip position.

Subsequently, the hip trajectory and the feet trajectories are fed into a generalized inverse kinematics algorithm, which yields a whole-body walking trajectory that can be applied directly on the robot.

3.2. Desired velocity and heading angle sequences

For the NMPC framework, the desired CoM velocities over the prediction horizon can be represented as two time-series vectors corresponding to the current time instant k . The desired CoM velocity sequences over the prediction horizon are defined as

$$\begin{aligned}\dot{\mathbf{X}}_{k+1}^{\text{ref}} &= [\dot{x}_{k+1}^{\text{ref}}, \dot{x}_{k+2}^{\text{ref}}, \dots, \dot{x}_{k+N}^{\text{ref}}]^T, \\ \dot{\mathbf{Y}}_{k+1}^{\text{ref}} &= [\dot{y}_{k+1}^{\text{ref}}, \dot{y}_{k+2}^{\text{ref}}, \dots, \dot{y}_{k+N}^{\text{ref}}]^T,\end{aligned}\quad (10)$$

where $\dot{x}_{k+j}^{\text{ref}}$ ($j = 1, 2, \dots, N$) is the desired CoM velocity in the world frame's x direction at the $k + j$ -th time step, and $\dot{y}_{k+j}^{\text{ref}}$ ($j = 1, 2, \dots, N$) is the corresponding velocity in y direction. These discrete velocity values are generated by assigning the reference velocity sequence (i.e., user-specified step-wise average desired velocities $\dot{\mathbf{X}}_{\text{step}}^{\text{ref}}$ and $\dot{\mathbf{Y}}_{\text{step}}^{\text{ref}}$, as detailed in Section 3.1) according to the NMPC sampling period T .

To ensure the robot's heading always aligns with its actual motion direction, the reference heading angle sequence $\boldsymbol{\theta}_{k+1}^{\text{ref}}$ is derived from the desired velocity sequences $\dot{\mathbf{X}}_{k+1}^{\text{ref}}$ and $\dot{\mathbf{Y}}_{k+1}^{\text{ref}}$. Defined over the same prediction horizon as the desired velocity, it is given by

$$\boldsymbol{\theta}_{k+1}^{\text{ref}} = [\theta_{k+1}^{\text{ref}}, \theta_{k+2}^{\text{ref}}, \dots, \theta_{k+N}^{\text{ref}}]^T, \quad (11)$$

where each element $\theta_{k+j}^{\text{ref}}$ (the reference heading angle at the $k + j$ -th time step) is calculated using the arctangent function with quadrant discrimination ($\arctan 2$) to avoid directional ambiguity. Specifically, it is defined as

$$\theta_{k+j}^{\text{ref}} = \arctan 2(\dot{y}_{k+j}^{\text{ref}}, \dot{x}_{k+j}^{\text{ref}}). \quad (12)$$

Notably, $\theta_{k+j}^{\text{ref}}$ represents the counterclockwise angle between the robot's reference heading and the x -axis of the world frame. Particularly when both $\dot{x}_{k+j}^{\text{ref}}$ and $\dot{y}_{k+j}^{\text{ref}}$ are zero (i.e., the user commands the robot to stop), $\theta_{k+j}^{\text{ref}}$ is kept consistent with the previous time step's reference heading $\theta_{k+j-1}^{\text{ref}}$ to avoid unnecessary heading fluctuations.

3.3. Objective function

The objective function used in the NMPC is given by

$$\min_{\mathbf{U}_k} \frac{\alpha}{2} J_1(\mathbf{U}_k) + \frac{\beta}{2} J_2(\mathbf{U}_k) + \frac{\gamma}{2} J_3(\mathbf{U}_k) + \frac{\epsilon}{2} J_4(\mathbf{U}_k), \quad (13)$$

where α, β, γ , and ϵ are non-negative weight coefficients that adjust the priority of each control objective, and $\mathbf{U}_k$ is the optimization variable vector defined as

$$\mathbf{U}_k = \begin{bmatrix} \ddot{\mathbf{X}}_k \\ \mathbf{X}_k^f \\ \ddot{\mathbf{Y}}_k \\ \mathbf{Y}_k^f \end{bmatrix}. \quad (14)$$

$J_1(\mathbf{U}_k)$ is the cost function for control input smoothing, defined as

$$J_1(\mathbf{U}_k) = \|\ddot{\mathbf{X}}_k\|^2 + \|\ddot{\mathbf{Y}}_k\|^2. \quad (15)$$

$J_2(\mathbf{U}_k)$ is the cost function for CoM velocity tracking, defined as

$$J_2(\mathbf{U}_k) = \|\dot{\mathbf{X}}_{k+1} - \dot{\mathbf{X}}_{k+1}^{\text{ref}}\|^2 + \|\dot{\mathbf{Y}}_{k+1} - \dot{\mathbf{Y}}_{k+1}^{\text{ref}}\|^2. \quad (16)$$

$J_3(\mathbf{U}_k)$ is the cost function for ZMP tracking. As indicated in [27], weakly weighted ZMP tracking facilitates faster and more robust responses to changes in system state or desired CoM speed. It is defined as

$$J_3(\mathbf{U}_k) = \|\mathbf{Z}_{k+1}^x - \mathbf{Z}_{k+1}^{x,\text{ref}}\|^2 + \|\mathbf{Z}_{k+1}^y - \mathbf{Z}_{k+1}^{y,\text{ref}}\|^2. \quad (17)$$

$J_4(\mathbf{U}_k)$ is the cost function for heading angle tracking, which minimizes deviation between predicted heading angle sequence $\boldsymbol{\theta}_{k+1}$ and reference sequence $\boldsymbol{\theta}_{k+1}^{\text{ref}}$, defined as

$$J_4(\mathbf{U}_k) = \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_{k+1}^{\text{ref}}\|^2, \quad (18)$$

where $\boldsymbol{\theta}_{k+1} = [\theta_{k+1}, \theta_{k+2}, \dots, \theta_{k+N}]^T$ and each element θ_{k+j} is calculated as:

$$\theta_{k+j} = \arctan 2(y_{k+j+2H} - y_{k+j}, x_{k+j+2H} - x_{k+j}). \quad (19)$$

Here, H denotes the number of discrete prediction time steps contained in one walking step. The heading angle is calculated using the CoM positions at steps k and $k + 2H$ (instead of $k + H$) to avoid fluctuations caused by the natural and inevitable sway of the CoM within a single step. Using two full prediction intervals smooths the heading estimate and aligns it with the robot's overall walking direction, which enhances interpretability and control stability. However, choosing $k + 3H$, by contrast, would unnecessarily increase heading tracking errors.

3.4. The constraints

In the NMPC framework, constraints are critical to ensuring locomotion's physical feasibility and balance. Their main purposes include confining CoP within the support region, as well as ensuring

proper footstep positions and footstep orientations on the ground to avoid leg collisions while satisfying the robot's physical limits. Details of each constraint will be introduced in the following parts.

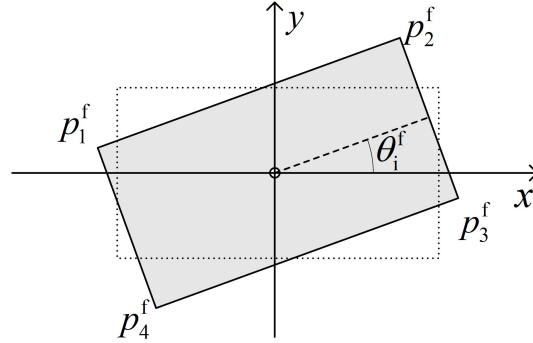


Figure 3. Shape of the support polygon (solid line) in the single support phase is a rectangle defined by vertices p_i^f . Its four edges are represented by $\mathbf{A}_{\text{cop}}(4 \times 2 \text{ matrix})$ and $\mathbf{B}_{\text{cop}}(4 \times 1 \text{ vector})$, with θ_i^f denoting the foot orientation.

3.4.1. Balance constraint

To maintain the humanoid robot's dynamic stability, the CoP must lie within the support polygon during both single support phases and double support phases.

In the single support phase, the support polygon is represented by $\mathbf{A}_{\text{cop}}$ and $\mathbf{B}_{\text{cop}}$, as shown in Figure 3. The CoP at instant k ($z_k = [z_k^x, z_k^y]^T$) must lie within this polygon, which can be expressed as

$$\mathbf{A}_{\text{cop}} \mathbf{R}(\theta_i^f)(z_k - \mathbf{f}_k) \leq \mathbf{B}_{\text{cop}}, \quad (20)$$

where $\mathbf{f}_k = [f_k^x, f_k^y]^T$ is the footstep position; $\theta_i^f = \theta_{k_i}$ denotes the orientation of the support foot which is equal to the heading angle of the i -th footstep at its placement instant k_i ; and $\mathbf{R}(\theta_i^f)$ is the rotation matrix defined as:

$$\mathbf{R}(\theta_i^f) = \begin{bmatrix} \cos \theta_i^f & -\sin \theta_i^f \\ \sin \theta_i^f & \cos \theta_i^f \end{bmatrix}. \quad (21)$$

For the double support phases, the support polygon is shown in Figure 4. The four vertices of each supporting foot (with $\eta \in \{L, R\}$ denoting the left or right foot, respectively) are calculated as:

$$\begin{bmatrix} p_{i,\eta}^x \\ p_{i,\eta}^y \end{bmatrix} = \begin{bmatrix} \cos \theta_i^f & \sin \theta_i^f \\ -\sin \theta_i^f & \cos \theta_i^f \end{bmatrix} \begin{bmatrix} \pm L/2 \\ \pm W/2 \end{bmatrix} + \begin{bmatrix} f_i^x \\ f_i^y \end{bmatrix}, \quad \eta \in \{L, R\}, \quad (22)$$

where $L/2$ and $W/2$ are half the foot's length and width, (f_i^x, f_i^y) represents the position of the footstep i , and θ_i^f is the orientation. The support polygon can be expressed approximately as a convex hexagon, whose vertices $\mathbf{p}_j = [p_j^x, p_j^y]^T$ ($j = 1, 2, \dots, 6$) are selected based on the upcoming footfall (left or right) and the desired heading angle. The CoP must lie inside this hexagon, enforced by the following set of inequalities:

$$\begin{cases} (p_{j+1}^x - p_j^x)(z_k^y - p_j^y) - (p_{j+1}^y - p_j^y)(z_k^x - p_j^x) \leq 0, & j = 1, 2, \dots, 5, \\ (p_{j-5}^x - p_j^x)(z_k^y - p_j^y) - (p_{j-5}^y - p_j^y)(z_k^x - p_j^x) \leq 0, & j = 6. \end{cases} \quad (23)$$

Under the geometric and attitude constraints presented in Sections 3.4.2 and 3.4.3, the hexagon shown in Figure 4 can effectively characterize the actual supporting polygon during the double-support phase, and this approximation method exhibits good universality. When the two feet are at extreme relative attitudes, area deviations will occur between the approximate model and the real supporting region. Numerical statistics demonstrate that working conditions with area deviations account for 19.62% of all cases within the constraint range. The maximum global relative area deviation of the model ((actual supporting area - hexagonal approximate area) / actual supporting area) stands at 18.5619%. This extreme value occurs under the limiting combination condition where the lateral projection step is 0.25 m and the attitude angle between the two feet is 30° .

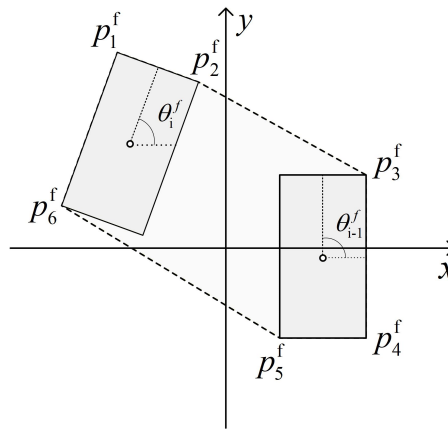


Figure 4. The support polygon is defined by six vertices $p_i^f = (p_i^x, p_i^y)$, indexed clockwise to form a closed convex region. It is formed by two supporting feet with orientations θ_i^f and θ_{i-1}^f .

3.4.2. Lateral footstep distance constraint

The lateral footstep distance constraint ensures kinematic feasibility and leg collision avoidance during locomotion by limiting the lateral span between two successive footsteps. The constraint is expressed as

$$S_{\min} \leq |S_i| \leq S_{\max}, \quad (24)$$

where $S_i = (f_{i+1} - f_i) \cdot u_i$ denotes the lateral distance between the i -th footstep, and $S_{\min}$ and $S_{\max}$ are the minimum and maximum allowable values of the lateral footstep distance, respectively.

3.4.3. Foot orientation constraint

One additional feasibility constraint considers the maximum and minimum allowable orientation angle change between adjacent steps to ensure the kinematic feasibility of the gait:

$$\Delta\theta_{\min} \leq |\theta_{i+1}^f - \theta_i^f| \leq \Delta\theta_{\max}, \quad (25)$$

where θ_i^f and θ_{i+1}^f denote the foot orientation angles at the i -th and $(i + 1)$ -th steps, respectively (as defined in 3.4.1). $\Delta\theta_{\min}$ and $\Delta\theta_{\max}$ are the minimum and maximum allowable orientation angle changes between two steps.

4. Simulation of turning movement

To verify the effectiveness of the proposed NMPC-based gait generation method for turning movement, simulations were conducted in the CoppeliaSim environment with the Bullet 2.78 physics engine and a fixed simulation time step of 4 ms was adopted. In this section, four sets of turning movement experiments were conducted with turning angles of 30°, 45°, 60°, and 90°, and all experiments were set to start turning from the 4th walking step. The simulation model of BHR-FCR was employed for these simulations.

Each walking step had a duration of 0.448 seconds, including 0.352 seconds of the single support phase and 0.096 seconds of the double support phase. Additionally, the preview horizon of the NMPC was configured as 4 full steps, ensuring a sufficient predictive scope for the unified optimization of the CoM trajectory, footstep sequence, and heading angle sequence. Moreover, the cost function gains were set as $\alpha = 10^{-12}$, $\beta = 10^{-8}$, $\gamma = 10^{-5}$, and $\epsilon = 10^{-7}$. As introduced in Section 3.3, α is for ensuring the smoothness of control inputs, β is for tracking the desired CoM velocity, γ is for the desired ZMP tracking, and ϵ is for tracking the desired heading angles. These gain values were selected based on the method's experimental performance in preliminary simulations.

Since the control input is the CoM jerk and the prediction horizon contains as many as 448 steps (four complete walking steps), the summation effect ensures that even a very small per-step weight yields a total cost of a reasonable magnitude. The reference heading angle θ^{ref} is computed from the reference velocities (Equation (12)). Therefore, the magnitude of ϵ is matched to that of β , with ϵ slightly larger to prioritise velocity tracking while avoiding abrupt heading fluctuations. Unlike conventional methods, the ZMP reference is not predefined but automatically constructed from the footstep sequence (Equation (7)). A large weight γ would force the CoP to closely follow an ideal trajectory, thereby suppressing footstep adaptation and impairing the robot's responsiveness to high speeds or large turning angles. Since the ZMP constraints themselves are already imposed as hard constraints (Equations (20) and (23)) that guarantee dynamic stability, γ is set to a relatively small value to serve only a guiding role.

The NMPC optimization problem is solved using the CasADi library in MATLAB on a computer equipped with an AMD Ryzen 7 5800 CPU (3.2 GHz). The solution time per planning update ranges from 100 ms to 200 ms, with fluctuations depending on the commanded walking speed and target turning angle. In the simulation, the NMPC iteratively optimizes the future CoM trajectory, footstep positions, and heading angles at the transition from single support phase to double support phase (i.e., at the moment a new step is initiated). Since the step duration is set to 448 ms in the simulation and real robot hardware usually adopts NMPC solvers implemented in C++, the actual computation latency is expected to be further reduced, making the proposed method fully compatible with real-time hardware requirements.

By specifying the reference velocity sequence $\dot{\mathbf{X}}_{\text{step}}^{\text{ref}}$ and $\dot{\mathbf{Y}}_{\text{step}}^{\text{ref}}$, the robot accelerates over the first two steps, with the magnitude of the resultant velocity stabilizing at 0.75 m/s starting from the 3rd step. The desired velocities are adjusted to initiate turning movements at the 4th step. As illustrated in Figure 5, the gait generation algorithm proposed in this study accurately tracked each desired turning angle at a relatively high resultant velocity.

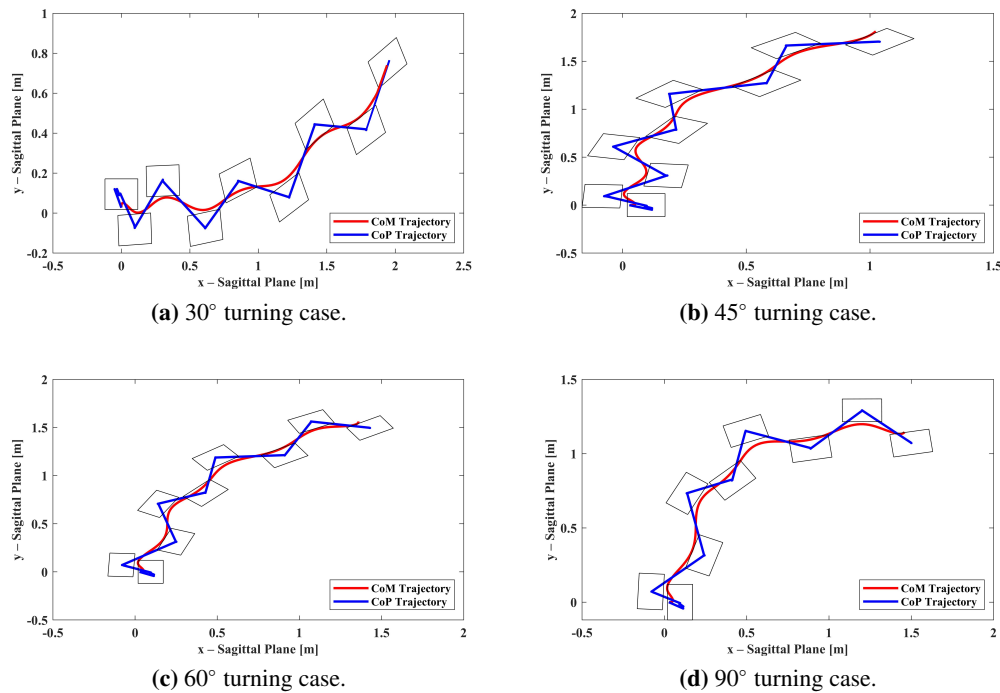


Figure 5. Footstep positions and orientations with CoP and CoM trajectories under different turning angles. The rectangles represent the planned foot support polygons, the blue line denotes the ZMP trajectory, and the red line represents the CoM trajectory.

Figure 6 compares the gaits generated by the existing method and the proposed method for the same 90° turning task. In the existing method, the double support phase is implicitly handled by simplifying it into a unified gait cycle, which forces the durations of the single support and double support phases to be strictly constrained by a fixed discrete time step. Also, this treatment cannot naturally exploit the transfer of the CoM or the subtle adjustments of foot orientation within the double support phase to assist the turning motion. As a result, the existing method struggles to realize large-angle turns efficiently. In contrast, the proposed method explicitly models the double support phase and incorporates the CoM transfer and foot orientation adjustments into the unified optimization framework, thereby enabling more agile and stable turning movements.

As depicted in Figure 7, the tracking performance of the resultant velocity magnitude is shown for the four turning angle cases. It can be observed that the actual resultant speed of the robot's CoM follows the predefined reference speed of 0.75 m/s. Consistent with the inherent traits of bipedal walking, the CoM velocity fluctuates slightly around the desired value. During the turning phase (corresponding to the time interval from 1.792 s to 2.688 s, i.e., the 4th to 6th walking steps when the turning is initiated and executed), an observable decrease in the resultant velocity is detected across all four cases. This velocity change is a direct consequence of the multiple physical and kinematic constraints elaborated in Section 3.4. Following the completion of the turning motion, the actual resultant velocity of the CoM promptly reverts to the vicinity of the reference speed and stabilizes. This indicates that the proposed NMPC-based gait planning method can effectively balance the trade-off between constraint satisfaction and velocity tracking performance.

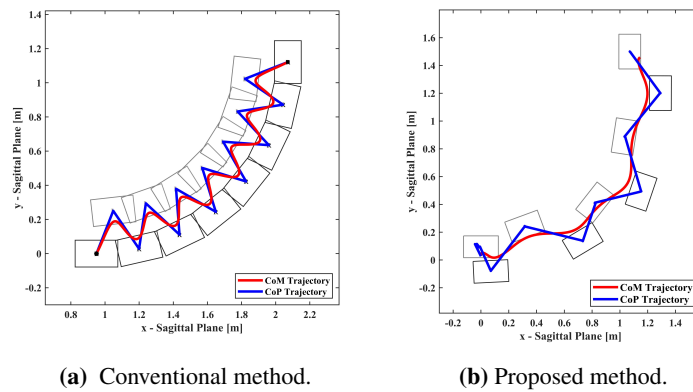


Figure 6. Comparison for 90° turning using the conventional method and proposed method. The rectangles represent the planned foot support polygons, the blue line denotes the ZMP trajectory, and the red line represents the CoM trajectory.

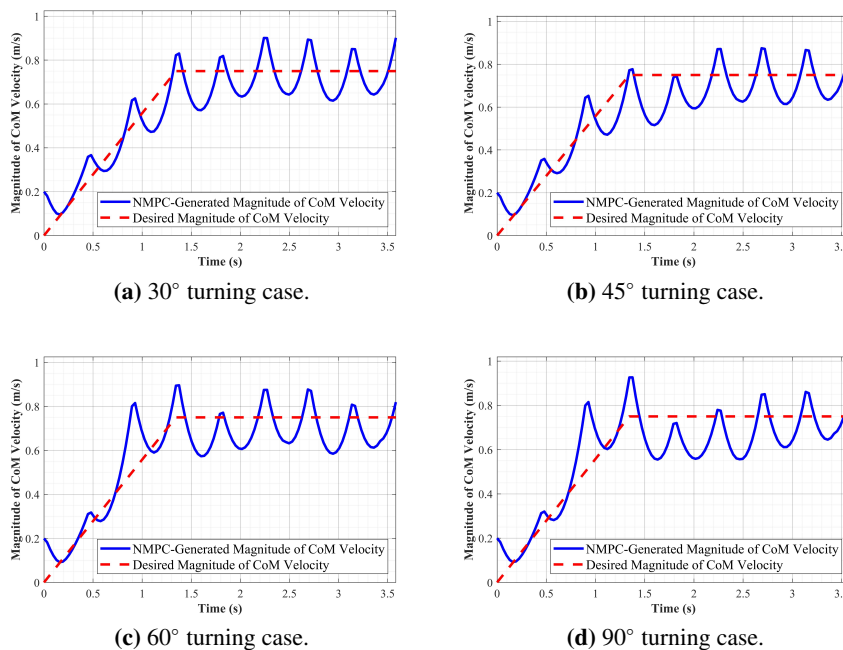


Figure 7. Tracking performance of the resultant velocity magnitude across different turning angles. The red dashed line denotes the user-defined reference speed, while the blue solid line represents the actual resultant speed of the CoM.

Figure 8 depicts the ZMP position errors in the x and y directions of the body frame at hip (a coordinate system established with the robot's hip as the origin) during the turning movements with target angles of 30°, 45°, 60°, and 90°, respectively. In the single support phase, the support polygon of the robot is a rectangle determined by the foot shape, with a length L of 0.09 m and a width W of 0.17 m; while in the double support phase, the support region is formed by the convex combination of the two feet's support polygons, presenting a larger feasible area compared to the single support phase. It

can be observed from the error curves that the ZMP position errors in both x and y directions remain within the bounds of the corresponding support regions throughout the entire movements.

To further quantify the locomotion performance, we evaluate the minimum ZMP stability margin in the X and Y directions, as well as the root mean square (RMS) velocity error under turning maneuvers of 30° , 45° , 60° , and 90° . During the tests, the minimum ZMP stability margin in the X direction is 0.05 m, 0.07 m, 0.06 m, and 0.05 m, and the corresponding values in the Y direction are 0.05 m, 0.05 m, 0.07 m, and 0.05 m. The RMS velocity errors are 0.08 m/s, 0.11 m/s, 0.12 m/s and 0.15 m/s. These quantitative results intuitively reflect the dynamic stability and velocity tracking accuracy of the robot during turning movements.

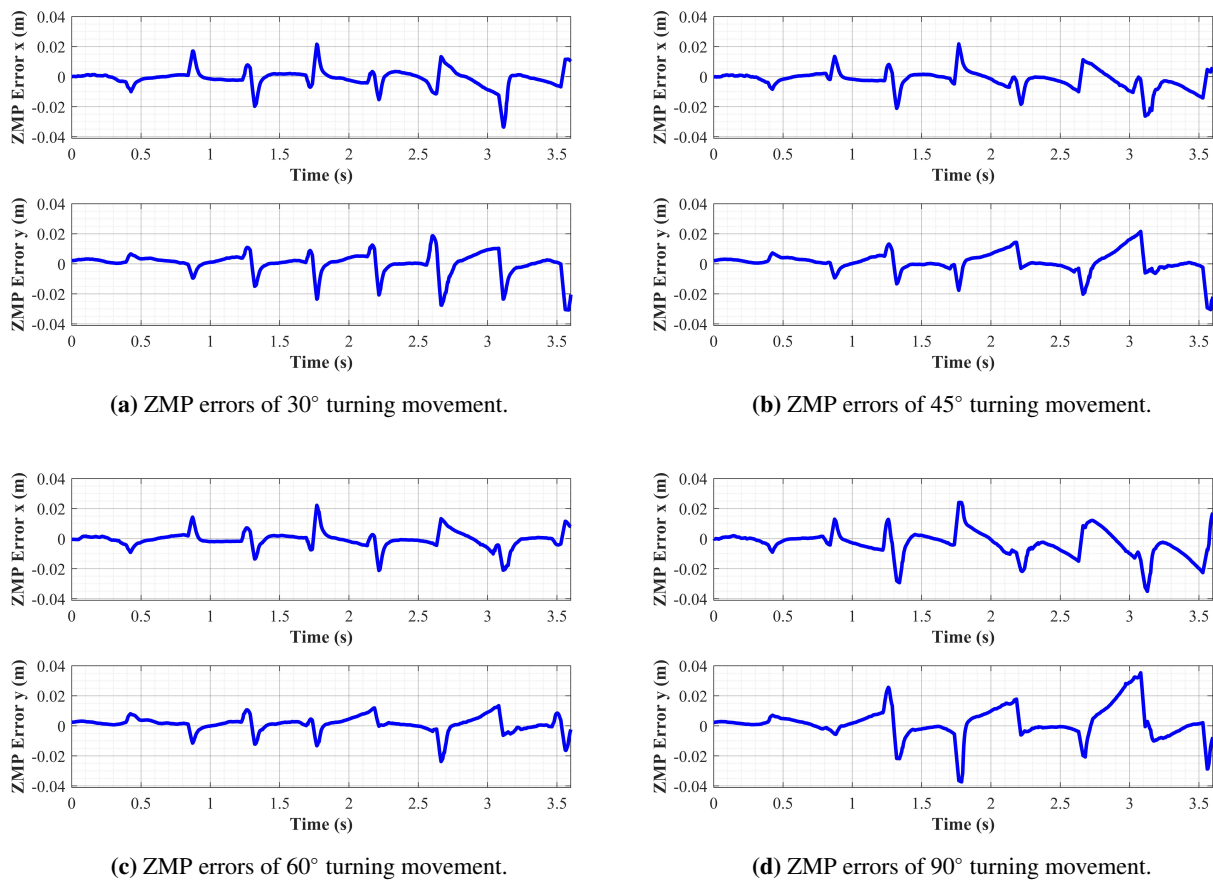


Figure 8. x/y -direction ZMP errors in body frame for different turning angles.

Figure 9 depicts the simulation snapshots of the BHR-FCR humanoid robot model performing the turning movements in the CoppeliaSim environment. To avoid potential collisions, the arms were set to maintain a fixed, slightly abducted posture throughout the experiments. From the sequential frames of the figure, the robot's turning process is clearly visualized. The robot starts from an initial straight walking posture as a simplification setting, and then gradually adjusts the positions and orientations of its footsteps to align with the trajectory planned by the algorithm. This confirms that the NMPC framework, which unifies the optimization of the CoM trajectory, footstep sequence, and the heading angle sequence, can reliably enable the robot to execute the designed turning movement.

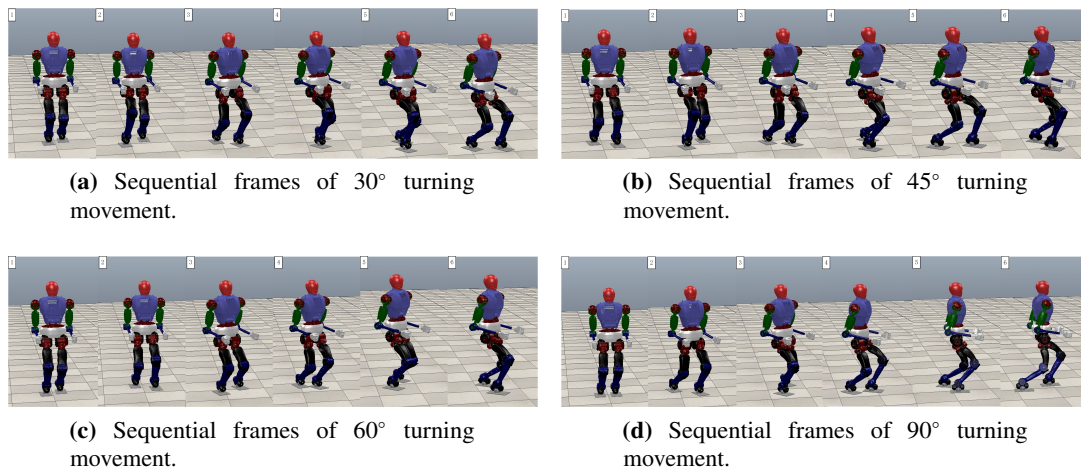


Figure 9. Sequential frames of the turning movement of the BHR-FCR humanoid robot in CoppeliaSim.

The results further demonstrate that the proposed method enables the humanoid robot to achieve stable and agile turning at a relatively high speed of 0.75 m/s, with all ZMP errors strictly confined within the support polygon. This capability to perform smooth, large-angle turns at high speed with fewer footsteps is practically meaningful, as it enhances the robot's maneuverability and adaptability in complex scenarios, such as disaster rescue and industrial inspection tasks, where flexible and stable turning is essential.

5. Conclusion

This paper proposes an NMPC-based gait generation algorithm for the turning movement of the humanoid robot. The heading angles are explicitly integrated into the NMPC framework, enabling the CoM trajectory, footstep positions, and heading angle to be optimized together and directly output in a single optimization process. Simulation experiments on the BHR-FCR humanoid robot model verify that the proposed algorithm effectively balances desired heading angles, desired velocity, and locomotion stability, enabling the robot to complete turning at a speed of 0.75 m/s with fewer footsteps.

6. Future works

Future research will extend the proposed NMPC-based gait planning method to complex terrains other than flat ground. Additionally, sequential quadratic programming (SQP) will be employed to decompose the NMPC problem into quadratic programming subproblems, aiming to reduce computational time. Inspired by the distributionally robust optimization-based stochastic self-triggered MPC framework for handling unbounded disturbances [28], and the distributionally robust MPC methods for state-feedback [29] and output-feedback [30] settings, future work could incorporate similar robust techniques to enhance the turning stability of humanoid robots under uncertain external perturbations. Furthermore, the arm swing dynamics and coupled angular momentum effects will be incorporated into the NMPC optimization framework to further improve the dynamic adaptability and control accuracy during high-speed turning motions.

The current work is limited to simulation validation without real-robot experiments. In practical deployment, the main challenges include additional control errors caused by dynamic modeling discrepancies between simulation and the real robot, as well as potential disturbances from complex and varying friction during high-speed turning motions in the real world.

Author contributions

Tian Mu: Conceptualization, Methodology, Investigation, Writing – original draft, Visualization. Junyao Gao: Supervision, Conceptualization, Writing – review & editing. Leilei Xie: Data curation, Validation. Mingyue Jin: Formal analysis, Software. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

During the preparation of this manuscript, the authors used ChatGPT for language polishing and expression improvement. All AI-assisted content was critically reviewed and revised by the authors, who take full responsibility for the final published version.

Acknowledgments

This work was supported in part by the Beijing Natural Science Foundation (Grant No. L243004).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, et al., The DARPA robotics challenge finals: Results and perspectives, *J. Field Robot*, **34** (2017), 229–240. <https://doi.org/10.1002/rob.21683>
2. A. Goswami, P. Vadakkepat, *Humanoid robotics: A reference*, Springer, 2019. https://doi.org/10.1007/978-94-007-6046-2_14
3. R. C. Luo, J. Sheng, C. C. Chen, P. H. Chang, C. I. Lin, Biped robot push and recovery using flywheel model based walking perturbation counteraction, *IEEE-RAS Int. Conf. Humanoid Robots*, 2013, 50–55. <https://doi.org/10.1109/HUMANOIDS.2013.7029954>
4. Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, Planning walking patterns for a biped robot, *IEEE Trans. Robot. Autom.*, **17** (2001), 280–289. <https://doi.org/10.1109/70.938385>
5. S. Kajita, T. Yamaura, A. Kobayashi, Dynamic walking control of a biped robot along a potential energy conserving orbit, *IEEE Trans. Robot. Autom.*, **8** (1992), 431–438. <https://doi.org/10.1109/70.149940>

6. Y. Cui, X. Fang, G. Liu, B. Li, Trajectory optimization of UAV based on hp-adaptive radar pseudospectral method, *J. Ind. Manag. Optim.*, **19** (2023), 1–18. <https://doi.org/10.3934/jimo.2021201>
7. Y. Zhang, Y. Yan, M. Tang, W. Wang, B. An, Trajectory tracking control of emergency supplies transport robots based on fuzzy MPC, *J. Ind. Manag. Optim.*, **19** (2023), 7616–7639. <https://doi.org/10.3934/jimo.2023011>
8. N. Scianca, M. Cagnetti, D. D. Simone, L. Lanari, G. Oriolo, Intrinsically stable MPC for humanoid gait generation, *IEEE-RAS Int. Conf. Humanoid Robots*, 2016, 601–606. <https://doi.org/10.1109/HUMANOIDS.2016.7803336>
9. S. Kajita, H. Hirukawa, K. Harada, K. Yokoi, *Introduction to humanoid robotics*, Springer, 2014. <https://doi.org/10.1007/978-3-642-54536-8>
10. M. Vukobratović, B. Borovac, Zero-moment point—Thirty five years of its life, *Int. J. Humanoid Robot.*, **1** (2004), 157–173. <https://doi.org/10.1142/s0219843605000466>
11. J. Pratt, J. Carff, S. Drakunov, A. Goswami, Capture point: A step toward humanoid push recovery, *IEEE-RAS Int. Conf. Humanoid Robots*, 2006, 200–207. <https://doi.org/10.1109/ICHR.2006.321385>
12. J. Reula, F. Canas, J. Pratt, A. Goswami, Learning capture points for humanoid push recovery, *IEEE-RAS Int. Conf. Humanoid Robots*, 2007, 65–72. <https://doi.org/10.1109/ICHR.2007.4813850>
13. J. Engelsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, G. Hirzinger, Bipedal walking control based on capture point dynamics, *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, 4420–4427. <https://doi.org/10.1109/IROS.2011.6048045>
14. J. Engelsberger, C. Ott, A. Albu-Schäffer, Three-dimensional bipedal walking control based on divergent component of motion, *IEEE Trans. Robot.*, **31** (2015), 355–368. <https://doi.org/10.1109/TRO.2015.2405592>
15. X. B. Peng, G. Berseth, K. Yin, M. V. D. Panne, DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning, *ACM Trans. Graph.*, **36** (2017), 1–13. <https://doi.org/10.1145/3072959.3073602>
16. Z. Xie, G. Berseth, P. Clary, J. Hurst, M. V. D. Panne, Feedback control for Cassie with deep reinforcement learning, *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, 1241–1246. <https://doi.org/10.1109/IROS.2018.8593722>
17. A. Foroutannia, M. R. Akbarzadeh-T, A. Akbarzadeh, A deep learning strategy for EMG-based joint position prediction in hip exoskeleton assistive robots, *Biomed. Signal Process. Control*, **75** (2022), 103557. <https://doi.org/10.1016/j.bspc.2022.103557>
18. A. Foroutannia, M. R. Akbarzadeh-T, A. Akbarzadeh, Adaptive fuzzy impedance control of exoskeleton robots with electromyography-based convolutional neural networks for human intended trajectory estimation, *Mechatronics*, **89** (2023), 102952. <https://doi.org/10.1016/j.mechatronics.2023.102952>

19. M. Shoryabi, A. Hajipour, A. Shoeibi, Recognition of anxiety and depression using gait data recorded by the Kinect sensor: a machine learning approach with data augmentation, *Sci. Rep.*, **15** (2025), 112345. <https://doi.org/10.1038/s41598-025-06535-3>
20. D. Rodriguez, S. Behnke, DeepWalk: Omnidirectional bipedal gait by deep reinforcement learning, *IEEE Int. Conf. Robot. Autom.*, 2021, 3033–3039. <https://doi.org/10.1109/ICRA48506.2021.9561717>
21. R. P. Singh, M. Benallegue, M. Morisawa, R. Cisneros, F. Kanehiro, Learning bipedal walking on planned footsteps for humanoid robots, *IEEE-RAS Int. Conf. Humanoid Robots*, 2022, 686–693. <https://doi.org/10.1109/Humanoids53995.2022.10000067>
22. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, Biped walking pattern generation by using preview control of zero-moment point, *IEEE Trans. Robot. Autom.*, **19** (2003), 186–194. <https://doi.org/10.1109/ROBOT.2003.1241826>
23. P. B. Wieber, Trajectory free linear model predictive control for stable walking in the presence of strong perturbations, *IEEE-RAS Int. Conf. Humanoid Robots*, 2006, 137–142. <https://doi.org/10.1109/ICHR.2006.321375>
24. P. B. Wieber, Viability and predictive control for safe locomotion, *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, 1–6. <https://doi.org/10.1109/IROS.2008.4651022>
25. M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, P. Souères, A reactive walking pattern generator based on nonlinear model predictive control, *IEEE Robot. Autom. Lett.*, **2** (2017), 10–17. <https://doi.org/10.1109/LRA.2016.2518739>
26. N. Scianca, D. D. Simone, L. Lanari, G. Oriolo, MPC for humanoid gait generation: Stability and feasibility, *IEEE Trans. Robot.*, **36** (2020), 1–18. <https://doi.org/10.1109/TRO.2019.2958483>
27. A. Herdt, H. Diedam, P. B. Wieber, D. Dimitrov, Online walking motion generation with automatic foot step placement, *Adv. Robot.*, **24** (2010), 719–737. <https://doi.org/10.1163/016918610X493552>
28. B. Li, W. Zhang, M. Shi, A distributionally robust optimization-based stochastic self-triggered model predictive control, *IEEE Trans. Syst. Man. Cybern. Syst.*, **56** (2026), 1–12. <https://doi.org/10.1109/TSMC.2025.3649661>
29. B. Li, Y. Tan, A. G. Wu, G. R. Duan, A distributionally robust optimization based method for stochastic model predictive control, *IEEE Trans. Autom. Control*, **67** (2022), 1–15. <https://doi.org/10.1109/TSMC.2025.3649661>
30. B. Li, T. Guan, L. Dai, G. R. Duan, Distributionally robust model predictive control with output feedback, *IEEE Trans. Autom. Control*, **69** (2024), 3270–3277. <https://doi.org/10.1109/TAC.2023.3321375>



AIMS Press

©2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)