*Robotica* (2026), **44**, pp. 1–36
doi:10.1017/S0263574725103020

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

# Limit cycles generation for energy-efficient bipedal walking via nonlinear model predictive control

Yuta Hanazawa, Yuhi Uchino and Shinichi Sagara

Department of Mechanical and Control Engineering, Kyushu Institute of Technology, Kitakyushu, Japan
**Corresponding author:** Yuta Hanazawa; Email: hanazawa@cntl.kyutech.ac.jp

**Received:** 21 January 2025; **Revised:** 20 October 2025; **Accepted:** 19 November 2025;
**First published online:** 23 December 2025

**Keywords:** biped robot; walking robot; model predictive control; limit cycle

## Abstract

This paper proposes an energy-efficient walking generation method utilizing limit cycles generated by nonlinear model predictive control (NMPC). Conventional limit cycle walking methods rely on strong feedback, such as output zeroing control, to attract the robot's state toward a predefined periodic trajectory. However, we hypothesize that employing feedback control that better leverages the robot's dynamics can improve energy efficiency during walking. Our previous work confirmed that using limit cycles generated by NMPC can produce energy-efficient walking patterns. This study builds upon this foundation and proposes a new method for generating walking in a general five-link bipedal robot. Through extensive numerical simulations, we demonstrate that the proposed method achieves highly energy-efficient walking while exhibiting excellent convergence to periodic trajectories.

## 1. Introduction

Research on bipedal robots has been conducted for decades, and recent advances in computing and artificial intelligence have accelerated progress in this field. While quadrupedal robots have also been widely studied, bipedal robots offer the distinct advantage of potentially serving as substitutes for humans in various tasks and environments [1–8]. Their ability to navigate human-accessible spaces makes them an exceptionally versatile form of mobility. However, bipedal robots are still under development, and fully functional humanoid robots capable of seamlessly replacing humans in complex tasks have not yet been realized. This makes the field a challenging yet rewarding area of research.

We primarily focused on developing gait generation methods that enable bipedal robots to achieve high-speed locomotion with superior energy efficiency while maintaining balance and preventing falls. Traditionally, zero-moment point (ZMP)-based walking has been the dominant approach for generating stable gaits in bipedal robots [9–11]. Moreover, advances in computational power and learning algorithms have led to a growing number of studies employing machine learning to achieve robotic walking. As a result, many robots capable of walking or running have been developed [12–16]. However, a key issue from the perspective of safety is the lack of mathematical justification for why such learned walking behaviors are stable and prevent the robot from falling. Limit cycle-based walking methods have emerged as a promising alternative, offering the potential for more natural, human-like walking patterns with enhanced energy efficiency. ZMP-based walking control has been widely adopted for its stability guarantees. Classical ZMP-based methods using the linear inverted pendulum (LIP) model impose the constraint that the ZMP must remain within the support polygon, which restricts the robot's motion during walking. Under the simplest LIP formulation with constant center of mass height, this constraint prohibits heel-strike and toe-off movements characteristic of human walking, requiring the entire foot to maintain contact with the ground as shown in Figure 1. Several extensions have been proposed to overcome these limitations while maintaining the ZMP framework. The variable height linear inverted

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

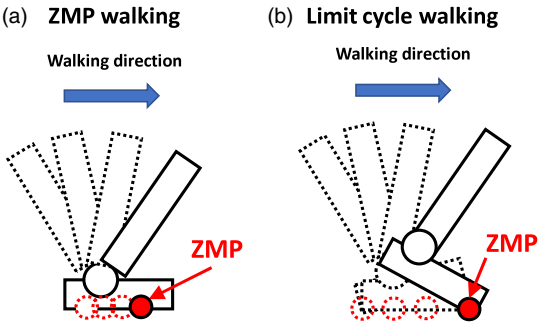



***Figure 1.*** *Variations in ground contact while walking.*

Pendulum (VHLIP) model allows height variation of the center of mass, enabling more natural trajectories while still utilizing ZMP constraints. The capture point (CP) concept [17, 18] and its extensions provide systematic methods for stability analysis and recovery. Additionally, approaches incorporating nonlinear dynamics and considering CoP (ZMP) constraints have been developed to achieve more natural motion [19].

Limit cycle-based walking does not impose the ZMP constraint and allows for more dynamic, underactuated behaviors such as heel-strike and toe-off, potentially leading to more efficient and natural gaits. While ZMP-based walking guarantees robust stability by explicitly maintaining the ZMP within the support polygon, limit cycle-based walking achieves stability by treating the walking motion as a stable limit cycle [20, 21]. Feedback control is then employed to drive the robot's state toward a predesigned periodic orbit. Although limit cycle-based walking does not guarantee stability under all conditions, it provides robustness against disturbances within the basin of attraction of the limit cycle. The ability to handle point contact walking itself is not novel. For example, the method in [19] employs hierarchical inverse dynamics to optimize joint accelerations and contact forces for momentum-based balance control, achieving stable and robust walking with planar feet. Since the formulation handles generalized contact forces, it could potentially be applied to point contacts with appropriate constraints. However, this method does not consider energy consumption and primarily aims to ensure stability and robustness. In contrast, our method can optimize energy consumption during walking online within a practical control cycle by minimizing control torques while ensuring stability as constraints.

Our proposed approach employs nonlinearmodel predictive control (NMPC) to determine joint torques online by minimizing the tracking error to the desired trajectory and the norm of joint torques during walking. The desired trajectory for energy-efficient walking can be designed offline in advance through optimization of Bézier polynomial coefficients. Although we do not perform this offline optimization in this paper, we demonstrate that such optimization is feasible using existing frameworks that explicitly consider limit cycle stability, such as [22]. This separation enables the robot to track the offline-designed energy-efficient trajectory while adapting to disturbances through predictive re-optimization, achieving efficient online computation (0.32 ms per cycle). As demonstrated by the simulations presented in this paper, this approach achieves energy-efficient walking with disturbance recovery capability without sacrificing computational efficiency.

Significant progress has been made in limit cycle-based walking methods, particularly those employing output zeroing control (OZC) through input–output linearization. Several research groups, including Grizzle's team, have utilized OZC to enhance convergence toward a predefined periodic orbit to achieve energy-efficient active walking [22–26]. Other groups have similarly applied this approach to realize stable limit cycle-based walking and running [2, 27–34]. Furthermore, in addition to OZC, both MABEL [24] and ATRIAS [30] improve energy efficiency of walking by incorporating large springs at the joints. These robots utilize target trajectories optimized to exploit their mechanical springs for energy-efficient walking, demonstrating effective utilization of the robots' passive dynamics. In this paper, we use the term "passive dynamics" to refer to the natural motion that emerges from the mechanical system without

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

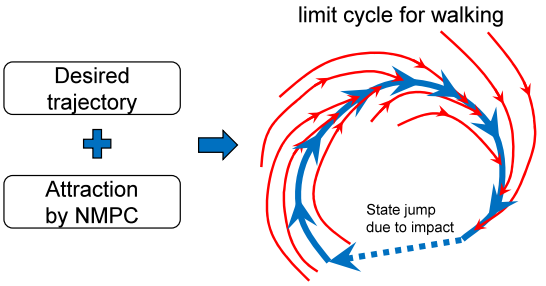



***Figure 2.*** *Limit cycle generation for biped walking.*

active control, which is determined by the system's inertia, gravity, and mechanical constraints. While the model used in this study does not incorporate springs for simplicity, the proposed approach is extensible to robots with elastic elements. Our proposed framework can optimize trajectories that leverage passive dynamics for improved energy efficiency, similar to these platforms. A key advantage of the NMPC-based approach proposed in this paper is its ability to maintain optimality even when the robot deviates from the desired trajectory. In typical OZC-based control, feedback linearization is employed to track a precomputed desired trajectory, whereas NMPC continuously re-optimizes the control input to guide the robot back to the desired trajectory while considering the full nonlinear dynamics.

OZC approach enables accurate tracking of a predesigned optimal trajectory under ideal conditions, resulting in energy-efficient walking. However, this method inherently relies on high-gain feedback to forcefully eliminate tracking errors without fully leveraging the robot's dynamics. In real-world environments, various disturbances such as uneven terrain, surface friction variations, motor backlash, and unexpected environmental contacts are inevitable. When such disturbances cause deviations from the desired trajectory, OZC tends to aggressively reconverge using high-gain control, which can result in excessive actuator torques, unwanted oscillations, and ultimately increased energy consumption and reduced stability. These limitations highlight the need for more practical control strategies that actively utilize the robot's dynamics and can respond to disturbances in a more flexible and energy-efficient manner. To address these challenges, we hypothesized that leveraging the robot's passive dynamics, similar to the concept of passive dynamic walking introduced by McGeer [35], could enable the generation of limit cycles with higher energy efficiency. By incorporating the robot's passive dynamics into the control strategy design, we aim to minimize energy losses caused by external disturbances and achieve more energy-efficient walking. Motivated by this idea, we previously proposed a gait generation method using NMPC to generate limit cycles [36]. This method uses NMPC to drive the robot's state toward a predesigned desired trajectory. NMPC predicts the robot's future states based on its current state and computes optimal control inputs by solving the optimization problem in real time.

The previously discussed concept of passive dynamic walking realizes energy-efficient locomotion by leveraging the nonlinear passive dynamics arising from the robot's mechanical structure and the gravitational field on a slope, thereby generating a stable limit cycle. In contrast, NMPC explicitly incorporates the nonlinear dynamics of the robot into a mathematical model, predicts its future behavior, and optimizes the control inputs to achieve convergence to a desired trajectory while minimizing energy consumption. In other words, whereas passive dynamic walking forms a limit cycle by utilizing external gravitational forces, NMPC aims to achieve similar natural dynamic behavior by applying only the minimal necessary actuation, thereby generating a limit cycle in a controlled manner. We have previously hypothesized that the NMPC-based approach can achieve energy-efficient walking similar to passive locomotion, even on level ground. This optimization considers the robot's dynamics and aims to minimize energy consumption while guiding the robot's state toward the desired trajectory, as illustrated in Figure 2. Although NMPC entails relatively high computational costs, recent advancements in algorithms for solving NMPC more efficiently [37, 38], coupled with the increasing performance of processors, suggest that its implementation on microcontroller-level systems will likely become feasible in the near future.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

In our previous work, we applied this method to a rimless wheel-type walking robot, a simplified model of bipedal locomotion, and demonstrated its effectiveness through numerical simulations. The results showed improved the energy efficiency of walking under ideal conditions, with particularly significant improvements observed when walking in disturbed environments [36]. However, since our ultimate goal is to achieve energy-efficient locomotion in general humanoid robots, the proposed method must be applicable to biped robots with a higher number of degrees of freedom (DoF). In typical multi-DoF bipedal robots, there exist large and rapid swing leg motions such as thigh swinging and knee bending that were not considered in the rimless wheel model. To enable future analytical optimization of limit cycle trajectories using hybrid zero dynamics (HZD) [22], it is essential that all robot states, including swing leg motions, converge strictly to the desired trajectories. Therefore, we aim to develop a control law for biped robots that includes swing leg dynamics and verify whether this control law can achieve both high energy efficiency and strict convergence to the desired walking trajectories.

In this study, we develop limit cycle walking control using NMPC that can be applied to a general and complex bipedal robot model, specifically a five-link robot with sagittal constraints. Numerical simulations were conducted to verify the effectiveness of the proposed method in generating energy-efficient and stable walking with high convergence to the desired trajectory. Furthermore, we implemented the proposed NMPC controller in C language (MEX on MATLAB) to evaluate its computational cost and demonstrate the feasibility of achieving real-time control within the constraints of a typical robot control cycle. The main contributions of this paper are as follows:

- Extended the NMPC-based walking control originally proposed for a simplified model [36] to a five-link robot by introducing virtual time parameterization within the prediction horizon. This novel extension enables the use of virtual constraints in the NMPC framework, demonstrating that the trajectory optimization framework of conventional HZD can be effectively integrated with NMPC.
- Demonstrated that the NMPC-based approach enables online re-optimization of control inputs considering full nonlinear dynamics, allowing the robot to maintain energy efficiency even when deviating from the desired trajectory due to disturbances or model uncertainties, unlike conventional HZD-based methods that rely on offline-optimized trajectories.
- Confirmed through numerical simulations that the proposed method generates motions satisfying virtual constraints, validating the integration of HZD and NMPC frameworks for the five-link robot model.
- Measured the computation time of the proposed method (0.32 ms per cycle), confirming its feasibility for real-time implementation within typical high-speed control cycles (1 ms), despite being approximately 12 times slower than conventional OZC (0.026 ms per cycle) due to online optimization.

The remainder of this paper is organized as follows: Section 2 introduces the dynamic model of the five-link bipedal robot. Section 3 details the proposed NMPC-based controller. Section 4 presents the results of our walking analysis and demonstrates the effectiveness of the proposed method through numerical simulations. Finally, Section 5 provides concluding remarks and discusses potential directions for future research.

## 2. Modeling of the five-link bipedal robot

### *2.1. Equation of motion*

We use a model of a five-link biped robot (Figure 3). Although we aim to apply our method to robots with flat feet, such as the illustrated in Figure 1, in future work, this paper focuses on a simplified case. Specifically, we consider a robot with point-contact feet that remain underactuated at all times. Figure 4 illustrates the details of the torques exerted by each motor between the links. The red arrows represent the torques generated by the motors, while the purple arrows indicate the reaction torques corresponding to those motor outputs. This robot also has four actuators for active control and its torso is equipped

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

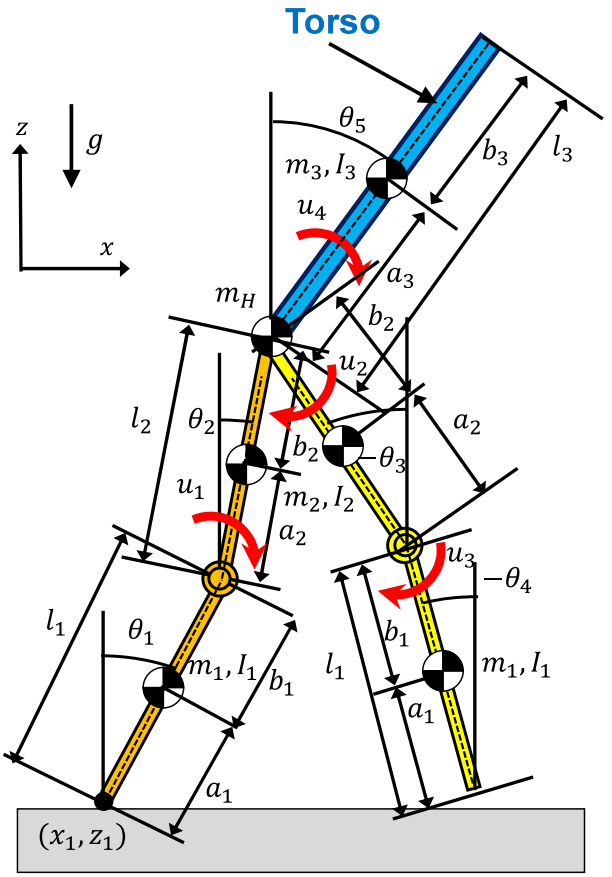



***Figure 3.*** *Five-link biped model.*

with a 9-axis sensor, which allows measurement of its orientation and angular velocity in the absolute coordinate frame. Each joint is equipped with an encoder that measures the relative joint angle between connected links. Angular velocities can then be obtained by differentiating these angle measurements. Therefore, we assume that the full state of the robot is observable. In this study, we derive the equations of motion using the Euler–Lagrange method, which is widely used in general applications. Equation of motion of the robot is given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{S}\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{q} = [\theta_1,\ \theta_2,\ \theta_3,\ \theta_4,\ \theta_5]^{\mathrm{T}}$ is the generalized coordinate vector, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{5\times5}$ is the inertia matrix, $\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^5$ is the vector that consists of Coriolis, centrifugal forces, as well as the gravitational vector, $\boldsymbol{u} = [u_1,\ u_2,\ u_3,\ u_4]^{\mathrm{T}}$ is the input torque vector, and $\boldsymbol{S} \in \mathbb{R}^{5\times4}$ is the driving matrix, which is expressed as:

$$\boldsymbol{S} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}.$$

The details of the equation of motion are provided in the Appendix.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

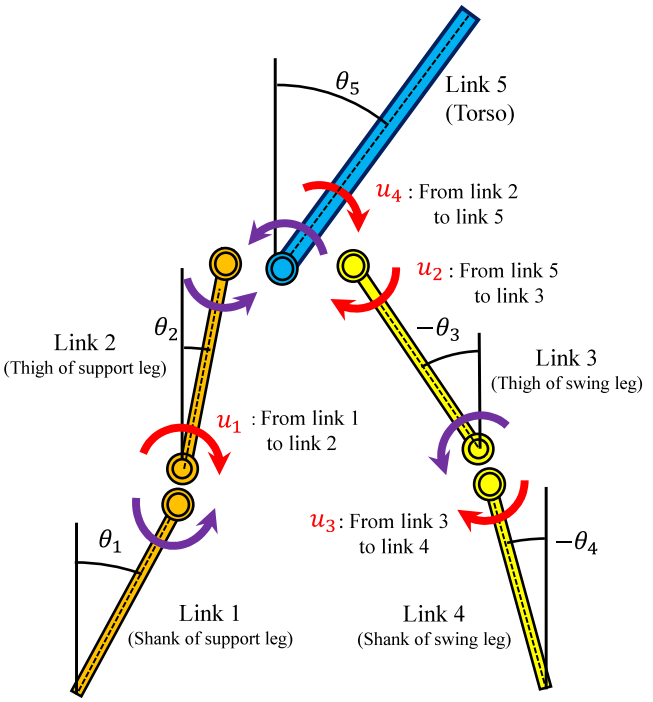



***Figure 4.*** *Details of the torques acting on the robot (the red arrows indicate the torques generated by the motors, while the purple arrows represent the reaction torques against the motor torques.).*

### *2.2. Impact equation*

The walking motion of a bipedal robot involves collision phenomena between the swing leg and the ground during locomotion, requiring the modeling of this behavior. To address this, a collision equation is introduced to compute the generalized velocities immediately after the collision, based on the robot's state immediately prior to impact. It is assumed that the collision between the swing leg and the ground is perfectly inelastic and instantaneous. At the moment of collision, velocity constraints arise in both the horizontal and vertical directions at the end of swing leg. These relationships can be expressed as follows:

$$l_1 \cos\theta_1\dot{\theta}_1 + l_2 \cos\theta_2\dot{\theta}_2 - l_2 \cos\theta_3\dot{\theta}_3 + l_1 \cos\theta_4\dot{\theta}_4 + \dot{x}_1 = 0,$$
$$-l_1 \sin\theta_1\dot{\theta}_1 - l_2 \sin\theta_2\dot{\theta}_2 + l_2 \sin\theta_3\dot{\theta}_3 + l_1 \sin\theta_4\dot{\theta}_4 + \dot{z}_1 = 0,$$

where $\dot{x}_1$ and $\dot{z}_1$ represent the velocities of the support leg's end in the horizontal and vertical directions, respectively. From the these equations, the instantaneous constraint Jacobian matrix at the moment of collision, $\boldsymbol{J}_I(\boldsymbol{q}_a) \in \mathbb{R}^{2\times 7}$, is derived as follows:

$$\boldsymbol{J}_I(\boldsymbol{q}_a) = \begin{bmatrix} l_1C_1 & l_2C_2 & -l_2C_3 & -l_1C_4 & 0 & 1 & 0 \\ -l_1S_1 & -l_2S_2 & l_2S_3 & l_1S_4 & 0 & 0 & 1 \end{bmatrix},$$

where $C_i$ is the $\cos\theta_i$ and $S_i$ is the $\sin\theta_i$. At this moment, the impulse vector $\boldsymbol{\lambda}_I \in \mathbb{R}^2$ generated during the collision and the generalized velocity of the robot immediately after the collision, $\dot{\boldsymbol{q}}_a^+ \in \mathbb{R}^7$, can be obtained as follows:

$$\boldsymbol{\lambda}_I = -\boldsymbol{X}_I(\boldsymbol{q}_a)^{-1}\boldsymbol{J}_I(q_a)\dot{\boldsymbol{q}}_a^-, \tag{2}$$
$$\boldsymbol{X}_I(\boldsymbol{q}_a) = \boldsymbol{J}_I(\boldsymbol{q}_a)\boldsymbol{M}_a(\boldsymbol{q}_a)^{-1}\boldsymbol{J}_I(\boldsymbol{q}_a)^{\mathrm{T}}, \tag{3}$$
$$\dot{\boldsymbol{q}}_a^+ = (\boldsymbol{I} - \boldsymbol{M}_a(\boldsymbol{q}_a)^{-1}\boldsymbol{J}_I(\boldsymbol{q}_a)^{\mathrm{T}}\boldsymbol{X}_I(\boldsymbol{q}_a)^{-1}\boldsymbol{J}_I(\boldsymbol{q}_a))\dot{\boldsymbol{q}}_a^-, \tag{4}$$

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

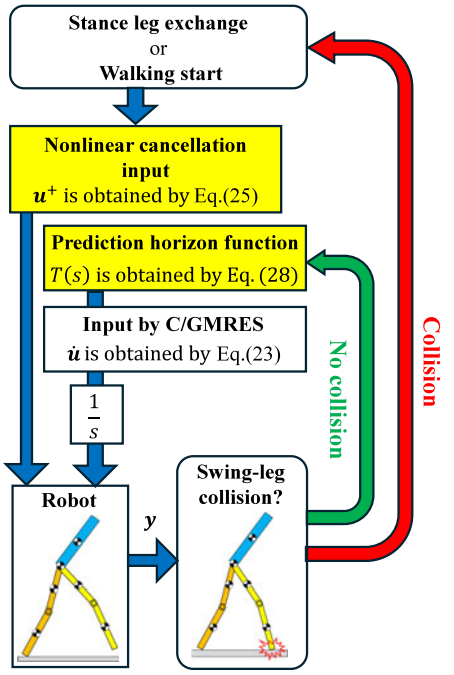



***Figure 5.*** *Control flow of the proposed NMPC: the yellow blocks represent the elements that were newly added in this paper to prevent divergence of the NMPC.*

where $\boldsymbol{q}_a = [\theta_1,\ \theta_2,\ \theta_3,\ \theta_4,\ \theta_5,\ x_1,\ z_1]^{\mathrm{T}}$ is the generalized coordinate vector in the augmented equation of motion and $\boldsymbol{M}_a(\boldsymbol{q}_a) \in \mathbb{R}^{7\times 7}$ represents the inertia matrix derived from the augmented equation of motion for the free-falling robot, incorporating the generalized coordinates and the coordinate components $x_1$ and $z_1$ of the support leg's end. Additionally, $\dot{\boldsymbol{q}}_a^- \in \mathbb{R}^7$ denotes the velocity vector immediately before the collision. Finally, the angular velocity obtained from Eq. (4) is used to update the robot's state vector to reflect its post-collision condition. The detailed derivation of the impact equations can be found in the literature by Grizzle et al. [23].

## 3. NMPC design for limit cycle walking

### *3.1. Overview of NMPC for biped robots*

In this paper, we construct NMPC for realizing walking motion of a five-link robot by incorporating several improvements based on the NMPC previously implemented on rimless wheels [36]. Figure 5 shows a schematic diagram of the control flow using the proposed method in this paper. The parts that significantly differ from prior research are indicated by yellow blocks. The main improvements over existing methods are the following three points. At the control steps immediately after gait initiation and stance leg exchange, the robot state changes discontinuously, and thus the control input by C/GMRES-based NMPC cannot be derived. Therefore, a separate input to mitigate the impact effect is employed at these steps. Second, we set the prediction horizon as a function dependent on virtual time. This is because the tracking performance to the desired trajectory of the swing leg motion, which changes rapidly during walking, was insufficient. Therefore, we resolve this issue by varying the prediction horizon according to virtual time. Finally, although not shown in this figure, we set the desired trajectory as a function dependent on virtual time and also modify the desired trajectory within the prediction by considering this virtual time. These details are presented below.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

### *3.2. Discrete-time NMPC*

This section describes the discrete-time NMPC applied to a biped walking robot. The discrete-time nonlinear system is defined as follows:

$$\boldsymbol{x}_{i+1}(t) = \boldsymbol{x}_i(t) + \boldsymbol{f}(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t))\Delta\tau, \tag{5}$$

$$\boldsymbol{x}_0(t) = \boldsymbol{x}(t), \tag{6}$$

where $i = 0, \ldots, N-1$, $N$ is the number of prediction steps, $\Delta\tau = T/N$ is the discrete interval of the prediction horizon, $T$ is the prediction horizon, and $\boldsymbol{x}_i(t) \in \mathbb{R}^n$ and $\boldsymbol{u}_i(t) \in \mathbb{R}^m$ are the state and input vectors at $i$-th step of the prediction horizon at time $t$, respectively. The objective function is defined as:

$$J := \varphi(\boldsymbol{x}_N(t)) + \Sigma_{i=0}^{N-1} L(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t))\Delta\tau, \tag{7}$$

where

$$\varphi(\boldsymbol{x}_N(t)) = \frac{1}{2}(\boldsymbol{x}_N(t) - \boldsymbol{x}_{d,N}(t))^{\mathrm{T}} \boldsymbol{S}_f (\boldsymbol{x}_N(t) - \boldsymbol{x}_{d,N}(t)), \tag{8}$$

$$L = \frac{1}{2}(\boldsymbol{x}_i(t) - \boldsymbol{x}_{d,i}(t))^{\mathrm{T}} \boldsymbol{Q} (\boldsymbol{x}_i(t) - \boldsymbol{x}_{d,i}(t)) + \boldsymbol{u}_i^{\mathrm{T}}(t) \boldsymbol{R} \boldsymbol{u}_i(t), \tag{9}$$

$\boldsymbol{x}_{d,i}$ is the desired state vector at $i$-th step of the prediction horizon, $\boldsymbol{S}_f$ is the weight for the terminal cost, $\boldsymbol{Q}$ is the weight for the stage cost of the state, and $\boldsymbol{R}$ is the weight for the stage cost of the input. Then, the Hamiltonian and necessary conditions for optimality are given obtained as:

$$H(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \boldsymbol{\lambda}_{i+1}(t)) := L + \boldsymbol{\lambda}_{i+1}^{\mathrm{T}}(t) \boldsymbol{f}(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t)), \tag{10}$$

$$\boldsymbol{\lambda}_N(t) = \frac{\partial \varphi^{\mathrm{T}}}{\partial \boldsymbol{x}}(\boldsymbol{x}_N(t)), \tag{11}$$

$$\boldsymbol{\lambda}_i(t) = \boldsymbol{\lambda}_{i+1}(t) + \frac{\partial H^{\mathrm{T}}}{\partial \boldsymbol{x}}(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \boldsymbol{\lambda}_{i+1}(t))\Delta\tau, \tag{12}$$

$$\frac{\partial H}{\partial \boldsymbol{u}}(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \boldsymbol{\lambda}_{i+1}(t)) = 0, \tag{13}$$

where $\boldsymbol{\lambda}_i(t) \in \mathbb{R}^n$ represents the adjoint (co-state) variable vector at $i$-th step. The optimal control input at the first step is used as the actual input for the system, followed by a receding of the finite horizon and repeating this sequence. To efficiently solve the NMPC, the C/GMRES method proposed by Ohtsuka [37] is employed. In this method, $\boldsymbol{U}(t)$ and $F(\boldsymbol{U}(t), \boldsymbol{x}(t), t)$ are defined as:

$$\boldsymbol{U}(t) := [u_0^{\mathrm{T}}(t), \ldots, u_{N-1}^{\mathrm{T}}(t)]^{\mathrm{T}}, \tag{14}$$

$$\boldsymbol{F}(\boldsymbol{U}(t), \boldsymbol{x}(t), t) := \begin{bmatrix} \boldsymbol{H}_u^{\mathrm{T}}(\boldsymbol{x}_0(t), \boldsymbol{u}_0(t), \boldsymbol{\lambda}_1(t)), \\ \vdots \\ \boldsymbol{H}_u^{\mathrm{T}}(\boldsymbol{x}_{N-1}(t), \boldsymbol{u}_{N-1}(t), \boldsymbol{\lambda}_N(t)), \end{bmatrix}, \tag{15}$$

where $H_u = \frac{\partial H}{\partial u}$. Subsequently, Eq. (10)–(13) are used to compute the state and adjoint variable vectors. By solving $\boldsymbol{F} = 0$ with the C/GMRES algorithm, $\dot{\boldsymbol{U}}(t)$ is obtained.

### *3.3. Extension of NMPC considering robustness*

To enhance robustness against modeling errors in NMPC and improve the stability of inputs when discrete state and input jumps occur, an augmented system is constructed by incorporating the input derivative $\dot{\boldsymbol{u}}$ as a new state. This approach, as proposed by Onodera et al. and Murayama et al. [39, 40],

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

treats the input as part of the state. Two types of discrete jumps must be considered: (a) jumps induced by minor collisions during walking (excluding swing leg collisions that trigger stance leg switching) and (b) jumps resulting from the control strategy in our approach, where the control input immediately after stance leg switching is designed to cancel the impact in the first step, followed by a transition to NMPC in subsequent steps. In both scenarios, the state–input relationship undergoes rapid changes, which can cause instability and potential divergence in NMPC. The incorporation of an integrator into the augmented system improves the numerical stability of NMPC even under these discrete jumps, thereby preventing optimization divergence and facilitating smoother walking motion.

The augmented system, which includes the input as a state for the five-link walking robot considered in this study, is expressed as:

$$\begin{aligned}\boldsymbol{x}(i+1) &= \begin{bmatrix} y_1(i+1) \\ y_2(i+1) \\ y_3(i+1) \\ y_4(i+1) \\ y_5(i+1) \\ y_6(i+1) \\ y_7(i+1) \\ y_8(i+1) \\ y_9(i+1) \\ y_{10}(i+1) \\ u_1(i+1) \\ u_2(i+1) \\ u_3(i+1) \\ u_4(i+1) \end{bmatrix} = \begin{bmatrix} y_1(i)+y_6(i)\Delta\tau \\ y_2(i)+y_7(i)\Delta\tau \\ y_3(i)+y_8(i)\Delta\tau \\ y_4(i)+y_9(i)\Delta\tau \\ y_5(i)+y_{10}(i)\Delta\tau \\ y_6(i)+\ddot{\theta}_1(i)\Delta\tau \\ y_7(i)+\ddot{\theta}_2(i)\Delta\tau \\ y_8(i)+\ddot{\theta}_3(i)\Delta\tau \\ y_9(i)+\ddot{\theta}_4(i)\Delta\tau \\ y_{10}(i)+\ddot{\theta}_5(i)\Delta\tau \\ u_1(i) \\ u_2(i) \\ u_3(i) \\ u_4(i) \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{10\times 4} \\ \boldsymbol{D} \end{bmatrix} \dot{\boldsymbol{u}}(i), \\ &= \boldsymbol{f}(\boldsymbol{x}(i)) + \boldsymbol{g}\dot{\boldsymbol{u}}(i), \end{aligned} \tag{16}$$

where $y_1(i), y_2(i), \ldots, y_5(i)$ are the output angles of the robot's links, $y_6(i), y_7(i), \ldots, y_{10}(i)$ are the output angular velocities of the robot's links, and $\dot{\boldsymbol{u}}(i) = [\dot{u}_1(i),\ \dot{u}_2(i),\ \dot{u}_3(i),\ \dot{u}_4(i)]^{\mathrm{T}}$ is the vector of the time derivative of the robot's inputs at $i$-th step of the prediction horizon. $\boldsymbol{D}$ is defined as:

$$\boldsymbol{D} = \begin{bmatrix} \Delta\tau & 0 & 0 & 0 \\ 0 & \Delta\tau & 0 & 0 \\ 0 & 0 & \Delta\tau & 0 \\ 0 & 0 & 0 & \Delta\tau \end{bmatrix}. \tag{17}$$

$\ddot{\boldsymbol{\theta}}(i) = [\ddot{\theta}_1(i),\ \ddot{\theta}_2(i),\ \ddot{\theta}_3(i),\ \ddot{\theta}_4(i),\ \ddot{\theta}_5(i)]^{\mathrm{T}}$ is the vector of the robot's angular acceleration at $i$-th step of the prediction horizon given by:

$$\ddot{\boldsymbol{\theta}}(i) = \boldsymbol{M}^{-1}(\boldsymbol{q}(i))(\boldsymbol{S}\boldsymbol{u}(i) - \boldsymbol{H}(\boldsymbol{q}(i), \dot{\boldsymbol{q}}(i))). \tag{18}$$

In this case, the objective function is defined as:

$$J := \varphi(\boldsymbol{x}(N), \boldsymbol{y}_d(N)) + \Sigma_{i=0}^{N-1} L(\boldsymbol{x}(i), \dot{\boldsymbol{u}}(i))\Delta\tau, \tag{19}$$

where

$$\varphi(\boldsymbol{x}(N), \boldsymbol{y}_d(N)) = \frac{1}{2}(\boldsymbol{y}(N) - \boldsymbol{y}_d(N))^{\mathrm{T}} \boldsymbol{S}_f (\boldsymbol{y}(N) - \boldsymbol{y}_d(N)), \tag{20}$$

$$\begin{aligned} L(\boldsymbol{x}(i), \dot{\boldsymbol{u}}(i), \boldsymbol{y}_d(i)) &= \frac{1}{2}(\boldsymbol{y}(i) - \boldsymbol{y}_d(i))^{\mathrm{T}} \boldsymbol{Q} (\boldsymbol{y}(i) - \boldsymbol{y}_d(i)) \\ &\quad + \frac{1}{2}\dot{\boldsymbol{u}}(i)^{\mathrm{T}} \boldsymbol{R}\dot{\boldsymbol{u}}(i), \end{aligned} \tag{21}$$

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

$\boldsymbol{y}_d(i)$ is the vector of the desired output values. The Hamiltonian is defined as:

$$H(\boldsymbol{x}(i), \dot{\boldsymbol{u}}(i), \boldsymbol{\lambda}(i+1)) := L + \boldsymbol{\lambda}^{\mathrm{T}}(i+1)(\boldsymbol{f}(\boldsymbol{x}(i)) + \boldsymbol{g}\dot{\boldsymbol{u}}(i)). \tag{22}$$

From Eq. (23), the necessary conditions for optimality are obtained as follows:

$$\frac{\partial H}{\partial \dot{\boldsymbol{u}}} = \dot{\boldsymbol{u}}^{\mathrm{T}}(i)\boldsymbol{R} + \boldsymbol{\lambda}^{\mathrm{T}}(i+1)\boldsymbol{g}, \tag{23}$$

$$= 0. \tag{24}$$

Therefore, the optimal control input $\dot{\boldsymbol{u}}$ can be obtained using C/GMRES, and the input vector $\boldsymbol{u}$ to be used in the real system can be obtained by integrating $\dot{\boldsymbol{u}}$.

Up to this point, the proposed method was an extension of our previous control law for a rimless wheel robot to a five-link biped [36]. However, this extension alone resulted in stability issues in the NMPC, particularly around the time of the swing leg impact. To address this issue, we further introduced improvements to the control input immediately after the impact (or at the beginning of the walking phase) and to the prediction horizon in the NMPC, as detailed below.

### *3.4. Nonlinear cancellation input*

For the first step at the start of walking and that immediately following a support leg transition, the input is not derived from model predictive control but rather calculated as follows:

$$\boldsymbol{u}^{+} = -(L_g L_f h)^{-1}(L_f^2 h - \boldsymbol{v}^{+}), \tag{25}$$

where $\boldsymbol{v}^{+} = [0,\ 0,\ 0,\ 0]^{\mathrm{T}}$. Leveraging the robot's dynamics, $L_g L_f h$ and $L_f^2 h$ are determined as follows:

$$L_g L_f h = \boldsymbol{C}\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{S}, \tag{26}$$

$$L_f^2 h = -\boldsymbol{C}\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{27}$$

and $\boldsymbol{C}$ is given by the following matrix:

$$\boldsymbol{C} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

This input is designed solely to cancel out gravitational, centrifugal, Coriolis, and inertial forces and does not contribute to correcting the robot's state toward the target trajectory. While this is not the optimal input, the control performance should not degrade significantly as long as the robot's state immediately after the support leg transition remains close to the target trajectory.

### *3.5. Design of prediction horizon function*

In NMPC, it is necessary to set a prediction horizon. In our NMPC, the prediction horizon is defined by the following function:

$$T(s) =: T_{\min} + \frac{(T_{\max} - T_{\min})(1 - \cos(2\pi s))}{2}, \tag{28}$$

where $T_{\min}$ and $T_{\max}$ are the minimum and maximum values of the prediction horizon, respectively. Then, $s$ is obtained as:

$$s(\theta_v) := \frac{\theta_v - \theta_v^{+}}{\theta_v^{-} - \theta_v^{+}}, \tag{29}$$

where $\theta_v = (\theta_1 + \theta_2)/2$ is the angle of the virtual support leg, and $\theta_v^{-}$ and $\theta_v^{+}$ are the angles immediately before and after the swing leg collision (after the support leg exchange), respectively. Here, $s$ is referred to as the virtual time, and it is a variable determined by the robot's state, independent of real time [41].

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

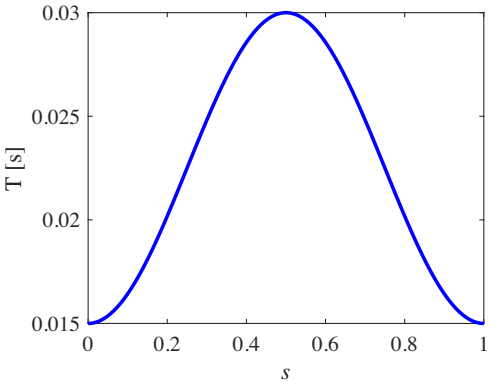



***Figure 6.*** *Prediction horizon with respect to $s(\theta_v)$ at $T_{\min} = 0.015$ s and $T_{\max} = 0.03$ s.*

The virtual time, $s$, is 0 immediately after the support leg transition, and it increases monotonically, reaching 1 just before the next support leg transition. Using this function, the prediction horizon changes as shown in Figure 6. Immediately after the support leg transition, the robot's dynamics change abruptly owing to leg collision; therefore, the prediction horizon should be kept short. However, during the middle of the stance phase, where the dynamics change more gradually, a longer prediction horizon is preferable. In the latter half of the stance phase, where swing leg collisions occur, the prediction horizon should be kept as short as possible. To meet these requirements, a prediction horizon function that depends on this virtual time was designed.

### *3.6. Design of desired trajectories*

Generating a stable limit cycle requires a periodic trajectory for the NMPC to track. This paper introduces a virtual constraint, represented as a Bézier curve, depending on a virtual time, as proposed by Westervelt et al. [22, 41]. The introduction of this virtual constraint ensures that the robot's motion is realized independently of real time. This approach simplifies the design of optimal trajectories for stable periodic walking. Furthermore, trajectory optimization can be performed within the framework of HZD [22]. For simplicity, we did not optimize the virtual constraint trajectory; instead, we determined it through a trial-and-error approach. Controlling the torso angle ($\theta_5$), relative angle of the support leg's knee joint ($\gamma$), relative angle between the support leg and the swing leg's hip joints ($\phi$), and angle of the swing leg's knee joint ($\eta$). The desired relative angle between the support and swing legs, $\phi_d(s)$ and that of the swing leg's knee joint, $\eta_d(s)$, are designed using Bézier curves with the virtual time parameter, $s$, as follows:

$$\phi_d(s) := \sum_{l=0}^{M} \alpha_l^{\phi} \frac{M!}{l!(M-l)!} s^l (1-s)^{M-l}, \tag{30}$$

$$\eta_d(s) := \sum_{l=0}^{M} \alpha_l^{\eta} \frac{M!}{l!(M-l)!} s^l (1-s)^{M-l}, \tag{31}$$

where $\alpha_l^{\phi}$ and $\alpha_l^{\eta}(s)$ are the control points that determine $\phi_d(s)$ and $\eta_d(s)$, respectively. The designed Bézier curves are shown in Figures 7 and 8. Moreover, the torso angle was set to $\theta_{5d} = 0.05$ rad and the knee angle of the support leg to $\gamma_d = 0.2$ rad. For simplicity, these two angles were assigned fixed values based on the assumption that they do not vary significantly during walking. However, it is also

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

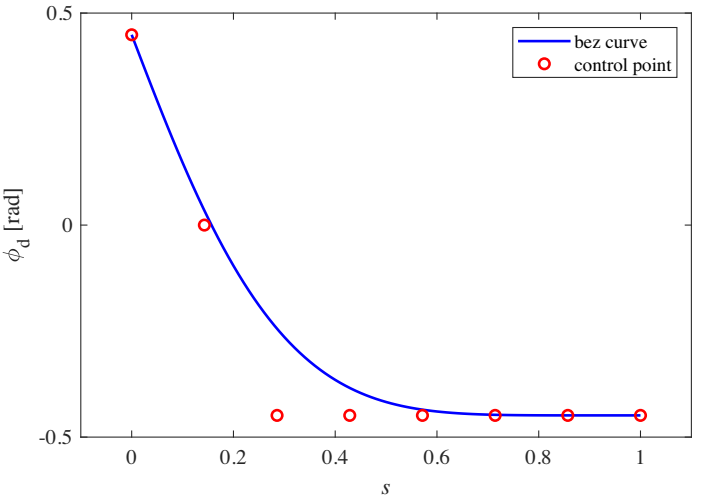



***Figure 7.*** *Desired trajectory of the thigh of swing leg at* $\alpha_0^{\phi} = \pi/7$, $\alpha_1^{\phi} = 0$, $\alpha_2^{\phi} = -\pi/7$, $\alpha_3^{\phi} = -\pi/7$, $\alpha_4^{\phi} = -\pi/7$, $\alpha_5^{\phi} = -\pi/7$, $\alpha_6^{\phi} = -\pi/7$, *and* $\alpha_7^{\phi} = -\pi/7$.

possible to define their continuous trajectories using Bézier curves. The details of $\boldsymbol{y}_d(i)$ are then obtained as follows:

$$\boldsymbol{y}_d(i) = \begin{bmatrix} 0 \\ y_1(i) - \gamma_d \\ y_2(i) + \phi_d(s) \\ y_3(i) + \eta_d(s) \\ \theta_{5d} \\ 0 \\ y_6(i) \\ y_7(i) + \dfrac{\partial \phi_d(s)}{\partial s} \dfrac{\partial s}{\partial \theta_v} \dfrac{\partial \theta_v}{\partial t} \\ y_8(i) + \dfrac{\partial \eta_d(s)}{\partial s} \dfrac{\partial s}{\partial \theta_v} \dfrac{\partial \theta_v}{\partial t} \\ 0 \end{bmatrix}. \tag{32}$$

## 4. Walking analysis

To demonstrate the effectiveness of the proposed method, its walking performance is compared with that of a conventional OZC approach through numerical simulations.

### *4.1. Details of analysis*

#### *4.1.1. OZC*

First, we briefly explain the OZC method employed in this study. The inputs and outputs are linearized, and the input obtained via proportional-derivative control is used as the new input. The input required

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

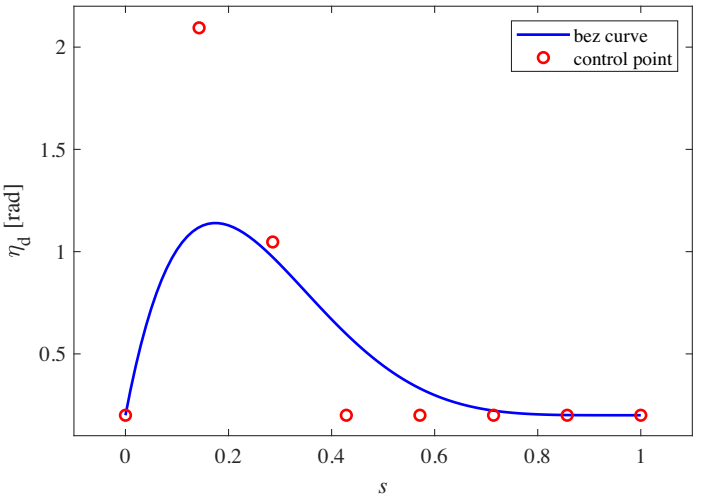



***Figure 8.*** *Desired trajectory of the knee of swing leg at* $\alpha_0^\eta = 0.2$, $\alpha_1^\eta = \pi/1.5$, $\alpha_2^\eta = -\pi/3$, $\alpha_3^\eta = 0.2$, $\alpha_4^\eta = 0.2$, $\alpha_5^\eta = 0.2$, $\alpha_6^\eta = 0.2$, *and* $\alpha_7^\eta = 0.2$.

for output zeroing is given by:

$$\boldsymbol{u} = -(L_g L_f h)^{-1}(L_f^2 h - \boldsymbol{v}), \tag{33}$$

where $\boldsymbol{v} = [v_1,\ v_2,\ v_3,\ v_4]^{\mathrm{T}}$ is the new input vector. The elements of $\boldsymbol{v}$ are then obtained as follows:

$$v_1 = -K_{P1}(\theta_2 - \theta_1 + \gamma_d) - K_{D1}(\dot{\theta}_2 - \dot{\theta}_1), \tag{34}$$

$$v_2 = -K_{P2}(\theta_3 - \theta_2 + \phi_d(s)) - K_{D2}\left(\dot{\theta}_3 - \dot{\theta}_2 + \frac{\partial \phi_d(s)}{\partial s}\frac{\partial s}{\partial \theta_v}\frac{\partial \theta_v}{\partial t}\right), \tag{35}$$

$$v_3 = -K_{P3}(\theta_4 - \theta_3 - \eta_d(s)) - K_{D3}\left(\dot{\theta}_4 - \dot{\theta}_3 - \frac{\partial \eta_d(s)}{\partial s}\frac{\partial s}{\partial \theta_v}\frac{\partial \theta_v}{\partial t}\right), \tag{36}$$

$$v_4 = -K_{P4}(\theta_5 - \theta_{5d}) - K_{D4}\dot{\theta}_5. \tag{37}$$

The gains were set as $K_{P1} = 6000$, $K_{D1} = 600$, $K_{P2} = 8000$, $K_{D2} = 800$, $K_{P3} = 8000$, $K_{D3} = 800$, $K_{P4} = 9000$, and $K_{D4} = 900$. These relatively high gains were set to enhance convergence to the desired trajectory.

### *4.1.2. NMPC*

The design parameters used in the NMPC are presented in Table I. The weights of NMPC are then set as follows. The terminal weights are set as $\boldsymbol{S}_f = \mathrm{diag}[0, 6\times10^6, 20\times10^6, 15\times10^6, 10\times10^6, 0, 11\times10^2, 85\times10, 5\times10^2, 8\times10^2, 0, 0, 0, 0]$. The state weights in the stage cost are set as $\boldsymbol{Q} = \mathrm{diag}[0, 45\times10^5, 18\times10^6, 13\times10^6, 96\times10^5, 0, 105\times10, 82\times10, 45\times10, 75\times10, 0, 0, 0, 0]$. The input weights in the stage cost are set as $\boldsymbol{R} = \mathrm{diag}[6\times10^{-2}, 15\times10^{-3}, 8\times10^{-2}, 2\times10^{-2}]$. These NMPC parameter settings need to be determined through a trial-and-error process; however, tuning some of the parameters may potentially benefit from the methods described in reference [42].

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

***Table I.*** *Control parameters for NMPC.*

| - | Symbol | Value | Unit |
|---|---|---|---|
| Max value of the prediction horizon | $T_{\max}$ | $3.0 \times 10^{-2}$ | s |
| Min value of the prediction horizon | $T_{\min}$ | $1.5 \times 10^{-2}$ | s |
| Number of divisions in the prediction horizon | $N$ | 4 | - |
| Sampling period (= control cycle) | $\Delta t$ | $1.0 \times 10^{-3}$ | s |
| Parameter for the continuous deformation method $\zeta$ | $1/\Delta t$ | - | |
| Max number of iterations | $k_{\max}$ | 4 | - |
| Time difference for forward differential approximation | $\delta t$ | $1.0 \times 10^{-8}$ | s |
| Allowable limit of error | $er$ | $1.0 \times 10^{-12}$ | - |

***Table II.*** *Physical parameters for the five-link biped robot.*

| Symbol | Unit | Value |
|---|---|---|
| $m_1$ | kg | 0.4 |
| $m_2$ | kg | 0.4 |
| $m_3$ | kg | 1.8 |
| $m_H$ | kg | 0.1 |
| $l_1$ | m | 0.4 |
| $l_2$ | m | 0.4 |
| $l_3$ | m | 0.6 |
| $a_1$ | m | $l_1/2$ |
| $a_2$ | m | $l_2/2$ |
| $a_3$ | m | $l_3/2$ |
| $b_1$ | m | $l_1/2$ |
| $b_2$ | m | $l_2/2$ |
| $b_3$ | m | $l_3/2$ |
| $I_1$ | kg·m$^2$ | $m_1 l_1^2/12$ |
| $I_2$ | kg·m$^2$ | $m_2 l_2^2/12$ |
| $I_3$ | kg·m$^2$ | $m_3 l_3^2/12$ |
| $g$ | m/s$^2$ | 9.81 |

#### *4.1.3. Simulation setup*

The physical parameters of the robot used in the numerical simulations are listed in Table II. The simulation comprises of MATLAB 2024a, Ryzen 9 9950X processor, 64 GB RAM, and Windows 11. The walking pattern established using these parameters is considered steady walking from the 15th step onward. The robot's walking speed and SR value, an index for evaluating energy efficiency, were calculated using walking data from the 15th to the 25th steps. The SR is defined as:

$$\mathrm{SR} := \frac{p}{Mgv_a}, \tag{38}$$

where $p$ [J/s] is the average input power, $M$ [kg] is the total weight of the robot, and $v_a$ [m/s] is the average walking speed of the robot. For the five-link bipedal robot used in this study, $p$ can be calculated using the following equation:

$$p := \frac{1}{T_t}\int_0^{T_t} (|u_1(\dot{\theta}_2 - \dot{\theta}_1)| + |u_2(\dot{\theta}_3 - \dot{\theta}_5)| + |u_3(\dot{\theta}_4 - \dot{\theta}_3)| + |u_4(\dot{\theta}_5 - \dot{\theta}_2)|)\mathrm{dt},$$

where $T_t$ [s] is the total walking time.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

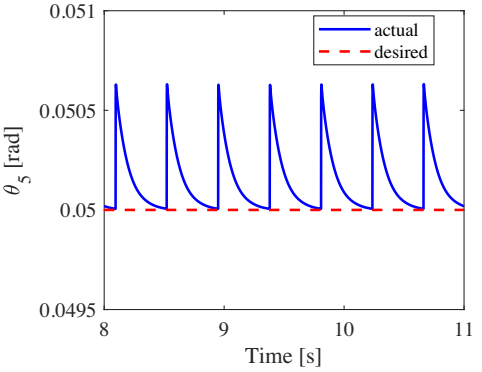



***Figure 9.*** *Variations in the torso angle during OZC-based walking with respect to time.*

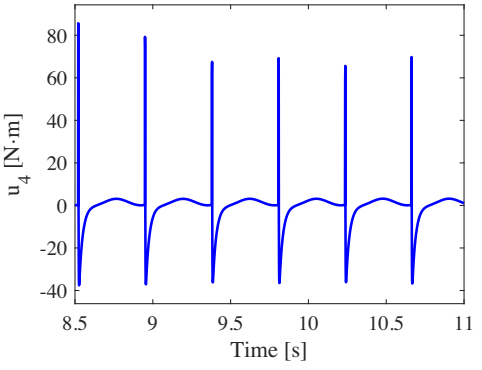



***Figure 10.*** *Variations in $u_4$ during OZC-based walking with respect to time.*

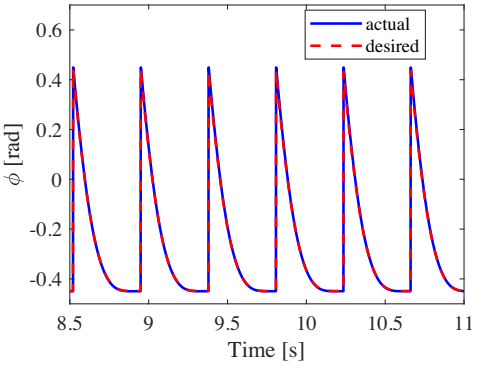



***Figure 11.*** *Variations in $\phi$ during OZC-based walking with respect to time.*

### *4.2. Walking under ideal environments*

#### *4.2.1. OZC-based walking*

First, the walking pattern using the conventional OZC method is presented. Figures 9–16 show the time-dependent changes in $\theta_5$, $u_4$, $\phi$, $u_2$, $\eta$, $u_3$, $\gamma$, and $u_1$ obtained during steady-state walking using the OZC-based approach in ideal environments, respectively. Evidently, the OZC method ensures that the robot strictly adheres to the desired trajectory. However, spike-like torques are observed at each input, occurring immediately after the support leg is switched. Figures 17 and 18 show the errors between

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

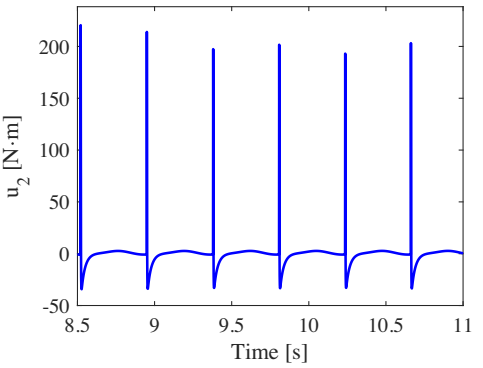



***Figure 12.*** *Variations in $u_2$ during OZC-based walking with respect to time.*

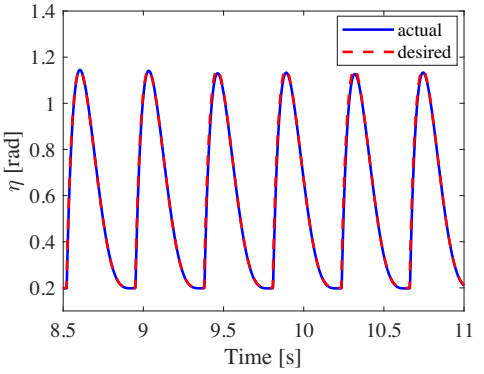



***Figure 13.*** *Variations in η during OZC-based walking with respect to time.*

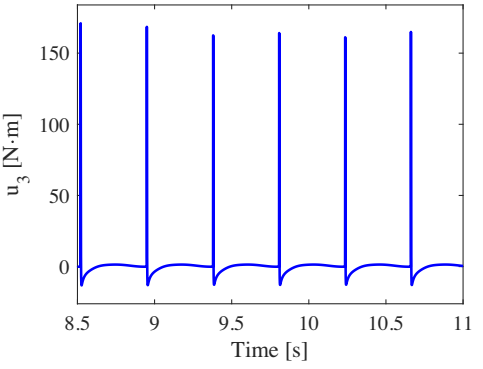



***Figure 14.*** *Variations in $u_3$ during OZC-based walking with respect to time.*

the desired values, $\phi_d$, $\eta_d$, and the actual values during steady-state walking using OZC, respectively. Immediately after the collision, the system deviates from the target trajectory but quickly converges back to it. Figures 19 and 20 show the phase diagrams of $\phi$-$\dot{\phi}$ and $\eta$-$\dot{\eta}$, respectively, during steady-state OZC walking. These figures confirm the formation of a limit cycle. The average walking speed with this OZC-based approach was 0.829 m/s, and the SR value was 1.442.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

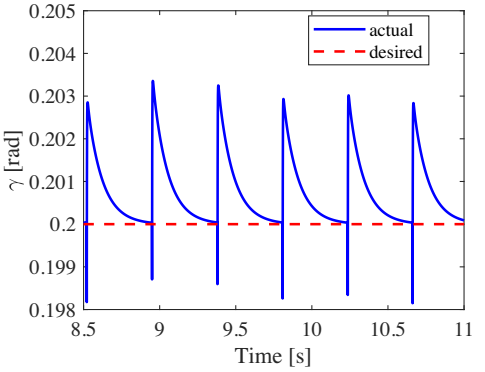



***Figure 15.*** *Variations in $\gamma$ during OZC-based walking with respect to time.*

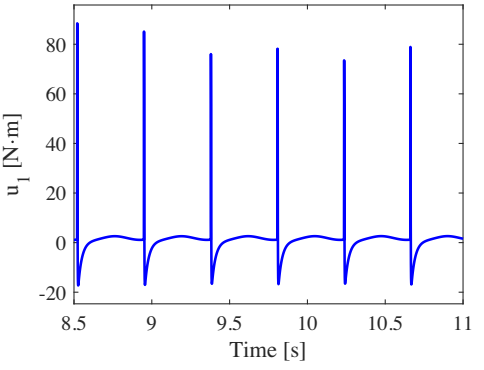



***Figure 16.*** *Variations in $u_1$ during OZC-based walking with respect to time.*

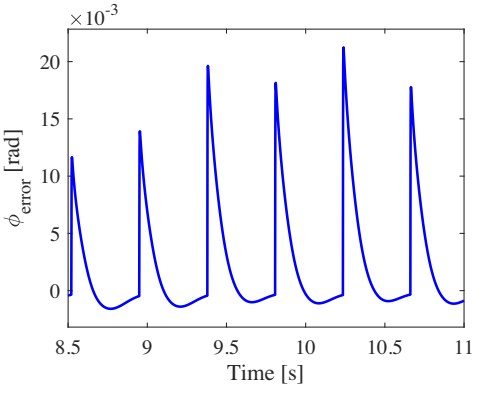



***Figure 17.*** *Variations in the error between the target value ($\phi_d$) and the actual value during OZC-based walking with respect to time.*

### *4.2.2. NMPC-based walking*

Subsequently, the walking performance using the proposed NMPC method was analyzed under the same conditions. Figures 21–28 show the time-dependent changes in $\theta_5$, $u_4$, $\phi$, $u_2$, $\eta$, $u_3$, $\gamma$, and $u_1$ during NMPC-based steady-state walking under ideal environments, respectively. Evidently, NMPC-based

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

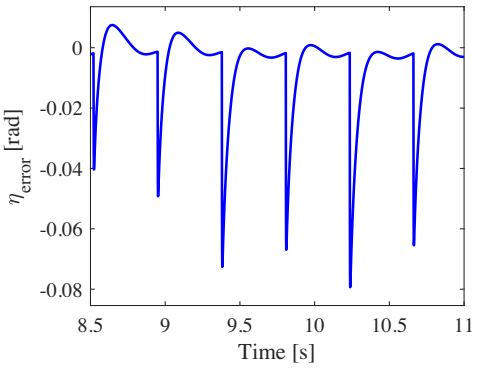



***Figure 18.*** *Variations in the error between the target value ($\eta_d$) and the actual value during OZC-based walking with respect to time.*

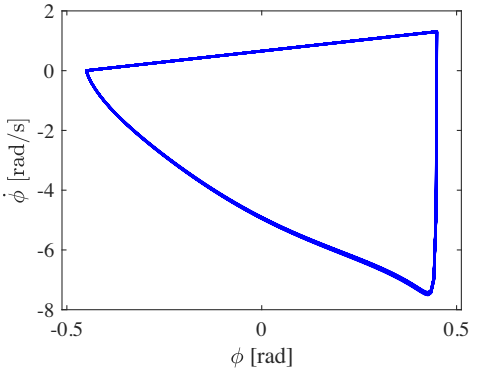



***Figure 19.*** *Phase diagram of $\phi$ and $\dot{\phi}$ during steady-state OZC-based walking.*

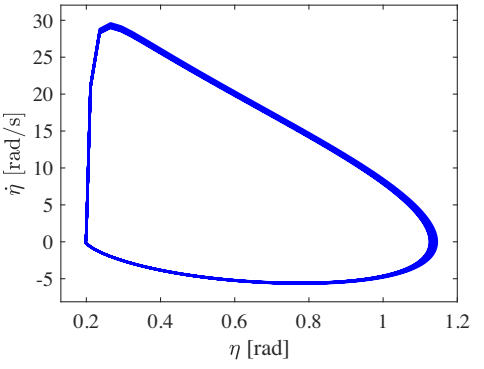



***Figure 20.*** *Phase diagram of $\eta$ and $\dot{\eta}$ during steady-state OZC-based walking.*

walking also converges to the desired trajectory, similar to OZC. However, unlike the walking achieved with OZC, no spike-like torques are generated, and the torque transitions smoothly.

Figures 29 and 30 show the errors between the target values, $\phi_d$, $\eta_d$, and the actual values during NMPC-based steady-state walking, respectively. As shown in these figures, the proposed NMPC-based

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

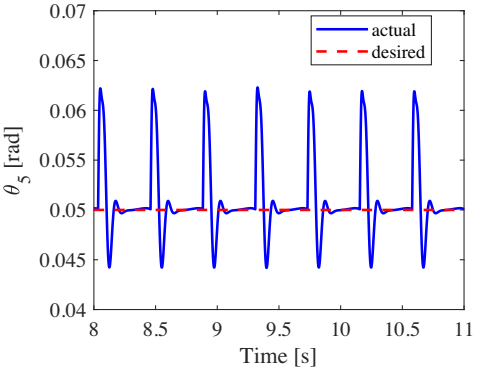



***Figure 21.*** *Variations in the torso angle during NMPC-based walking with respect to time.*

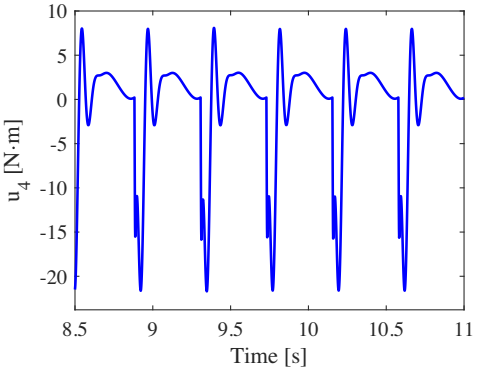



***Figure 22.*** *Variations in $u_4$ during NMPC-based walking with respect to time.*

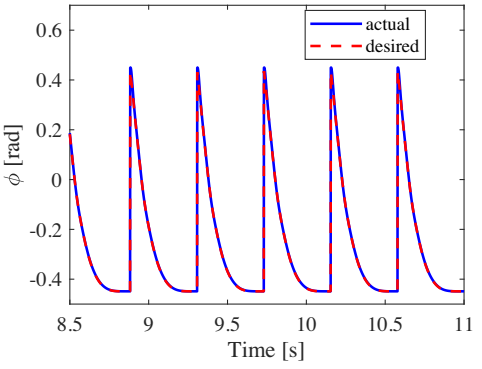



***Figure 23.*** *Variations in $\phi$ during NMPC-based walking with respect to time.*

walking demonstrates trajectory-tracking performance comparable to that with high-gain OZC, suggesting the potential feasibility of HZD-based trajectory optimization. Figures 31 and 32 show the phase diagrams of $\phi$-$\dot{\phi}$ and $\eta$-$\dot{\eta}$, during NMPC-based steady-state walking, respectively. These figures indicate that a limit cycle is also formed during NMPC-based walking. The average walking speed with the NMPC-based method was 0.832 m/s, and the SR value was 1.177. Thus, the proposed walking method

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

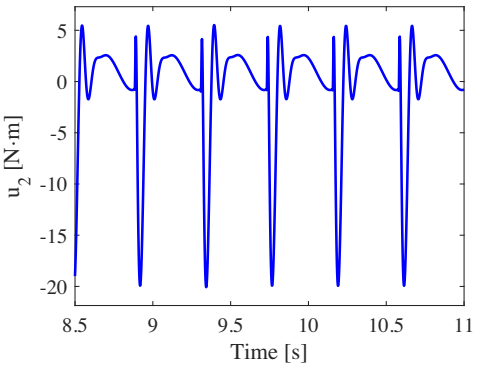



***Figure 24.*** *Variations in $u_2$ during NMPC-based walking with respect to time.*

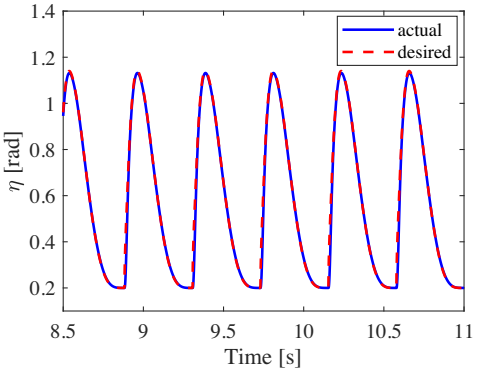



***Figure 25.*** *Variations in η during NMPC-based walking with respect to time.*

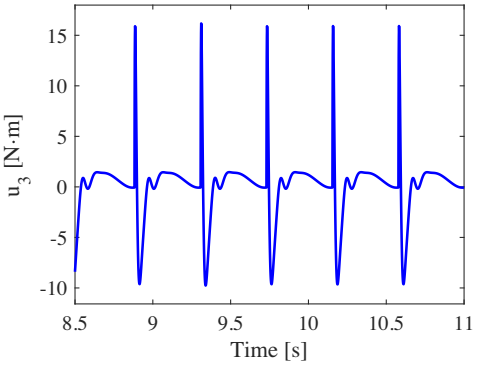



***Figure 26.*** *Variations in $u_3$ during NMPC-based walking with respect to time.*

achieves a similar speed and convergence to the desired trajectory to OZC-based walking approach, while significantly improving energy efficiency.

### *4.3. Walking in an environment with disturbances*

#### *4.3.1. OZC-based walking in disturbances*

The previous analysis examined walking under ideal conditions without disturbances. This section evaluates walking performance in an environment with disturbances, which were modeled as random torques

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

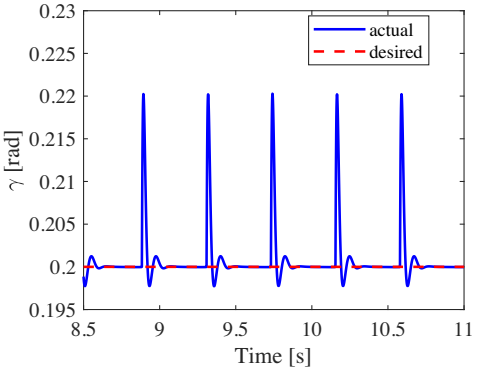



***Figure 27.*** *Variations in $\gamma$ during NMPC-based walking with respect to time.*

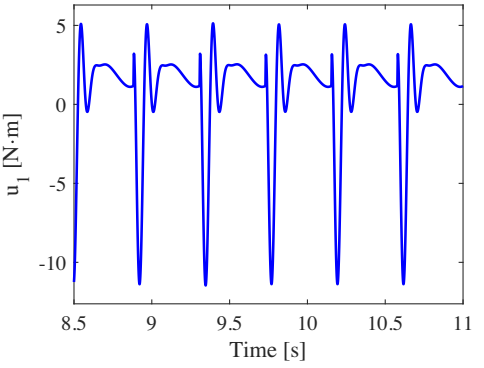



***Figure 28.*** *Variations in $u_1$ during NMPC-based walking with respect to time.*

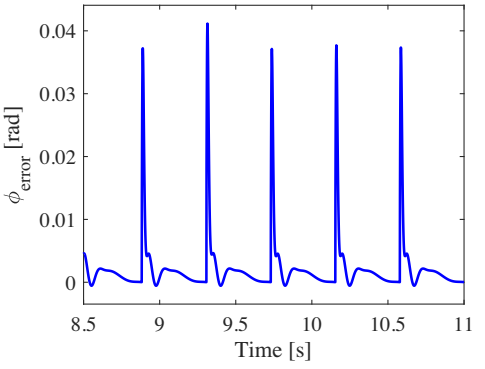



***Figure 29.*** *Variations in the error between the desired value ($\phi_d$) and the actual value during NMPC-based walking with respect to time.*

following a normal distribution (with the disturbance torque defined as 24 times the result of MATLAB's randn function), applied to each joint for 1 ms every 50 ms. Figures 33–40 show the time-dependent changes in $\theta_5$, $u_4$, $\phi$, $u_2$, $\eta$, $u_3$, $\gamma$, and $u_1$ during OZC-based steady-state walking in an environment with disturbances, respectively. Figures 41 and 42 show the errors between the target values, $\phi_d$, $\eta_d$, and

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

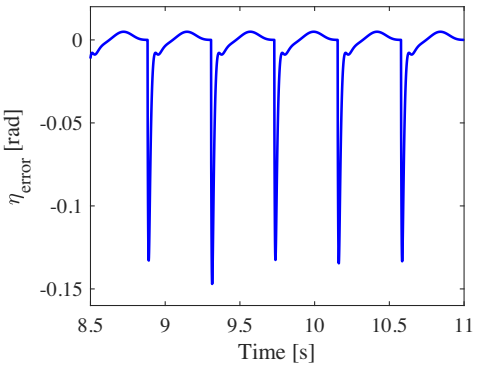



***Figure 30.*** *Variations in the error between the target value ($\eta_d$) and the actual value during NMPC-based walking with respect to time.*

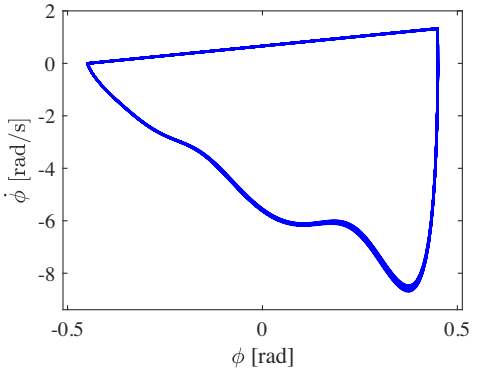



***Figure 31.*** *Phase diagram of $\phi$ and $\dot{\phi}$ during NMPC-based steady-state walking.*

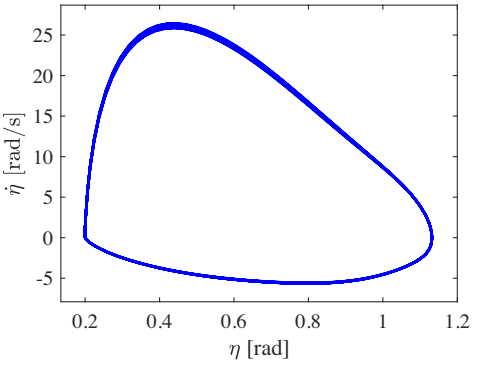



***Figure 32.*** *Phase diagram of $\eta$ and $\dot{\eta}$ during NMPC-based steady-state walking.*

the actual values during OZC-based steady-state walking in an environment with disturbances. Even in the presence of disturbances, high-gain OZC ensures strong tracking performance with the desired trajectory. However, the torque waveforms exhibit significant spikes, which are expected to considerably impact energy efficiency. Additionally, generating such large torques in practice is challenging, raising

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

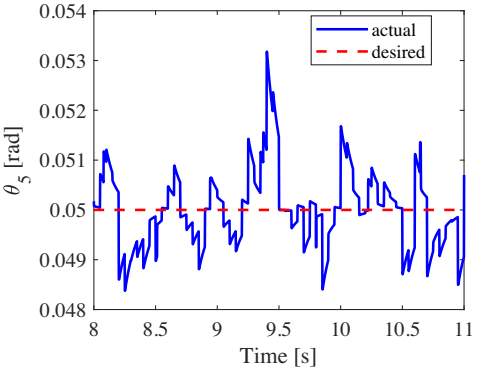



***Figure 33.*** *Variations in the torso angle during OZC-based walking in disturbance environment with respect to time.*

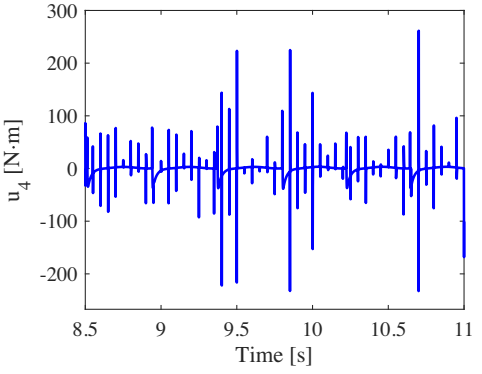



***Figure 34.*** *Variations in $u_4$ during OZC-based walking in disturbance environment with respect to time.*

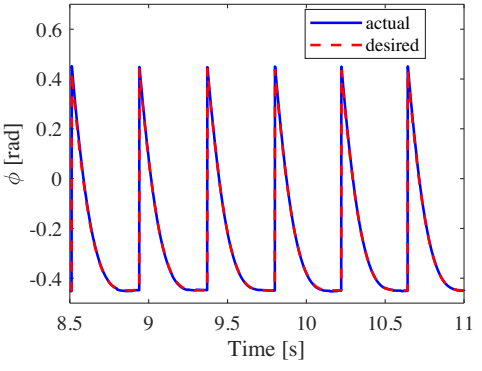



***Figure 35.*** *Variations in $\phi$ during OZC-based walking in disturbance environment with respect to time.*

concerns about the feasibility of this approach. Figures 43 and 44 show the phase diagrams of $\phi$-$\dot{\phi}$ and $\eta$-$\dot{\eta}$, during OZC-based steady-state walking under disturbances, respectively. Although the disturbances cause the robot's state to deviate from the limit cycle, it quickly converges back to the original trajectory. The average walking speed for the OZC-based approach under disturbances was 0.829 m/s and the SR value was 1.979. While the walking speed is almost the same as when there are no disturbances, energy consumption increased substantially, leading to a significant reduction in energy efficiency.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

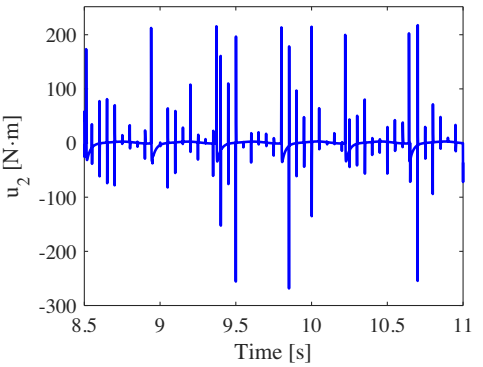



***Figure 36.*** *Variations in $u_2$ during OZC-based walking in disturbance environment with respect to time.*

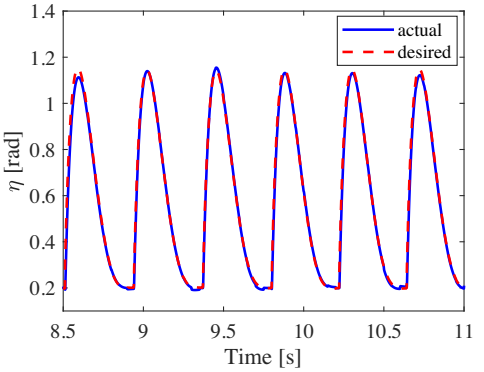



***Figure 37.*** *Variations in $\eta$ during OZC-based walking in disturbance environment with respect to time.*

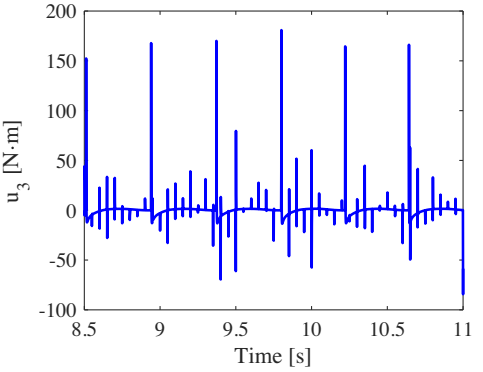



***Figure 38.*** *Variations in $u_3$ during OZC-based walking in disturbance environment with respect to time.*

### *4.3.2. NMPC-based walking in disturbances*

Figures 45–52 show the time-dependent changes in $\theta_5$, $u_4$, $\phi$, $u_2$, $\eta$, $u_3$, $\gamma$, and $u_1$ during NMPC-based steady-state walking in the disturbance environment, respectively. Figures 53 and 54 show the errors between the desired values, $\phi_d$, $\eta_d$, and the actual values during NMPC-based steady-state walking in the disturbance environment respectively. The proposed NMPC-based control achieves a level of convergence to the desired trajectory comparable to high-gain OZC. Additionally, no sharp torque spikes are observed, and the maximum torque values remain within realistic limits. Figures 55 and 56 show the

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

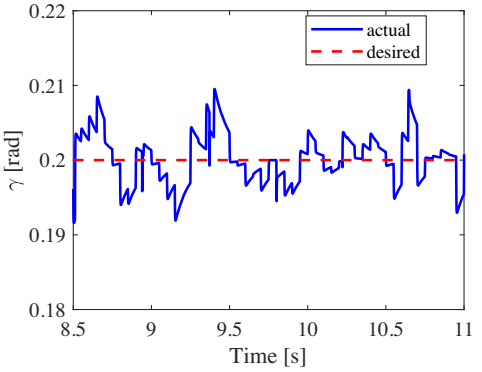



***Figure 39.*** *Variations in $\gamma$ during OZC-based walking in disturbance environment with respect to time.*

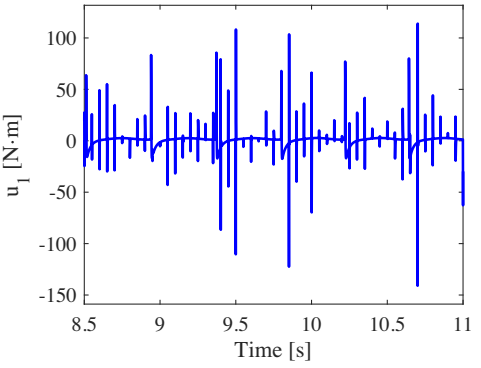



***Figure 40.*** *Variations in $u_1$ during OZC-based walking in disturbance environment with respect to time.*

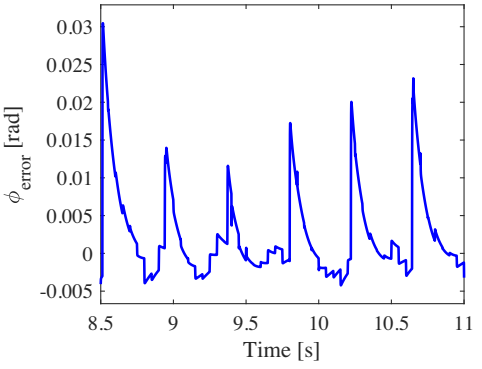



***Figure 41.*** *Variations in the error between the desired value ($\phi_d$) and the actual value during OZC-based walking in disturbance environment with respect to time.*

phase diagrams for $\phi$-$\dot{\phi}$ and $\eta$-$\dot{\eta}$ during NMPC-based steady-state walking under disturbances, respectively. Although the disturbances cause the robot's state to deviate from the trajectory, the state gradually converges back to the original trajectory, unlike in the case of OZC-based walking. The average walking speed using the NMPC-based control was 0.834 m/s and the SR value was 1.287. In conventional OZC-based walking, energy efficiency significantly deteriorates in disturbed environments. However, the proposed NMPC-based walking substantially mitigates this issue.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

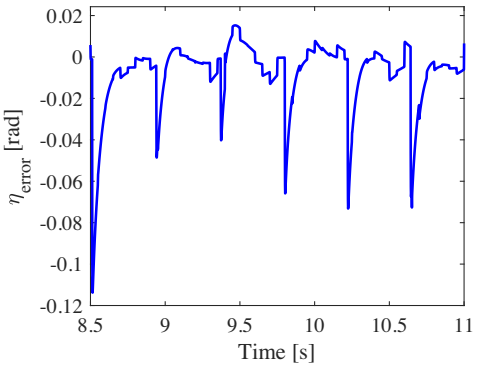



***Figure 42.*** *Variations in the error between the target value ($\eta_d$) and the actual value during OZC-based walking in disturbance environment with respect to time.*

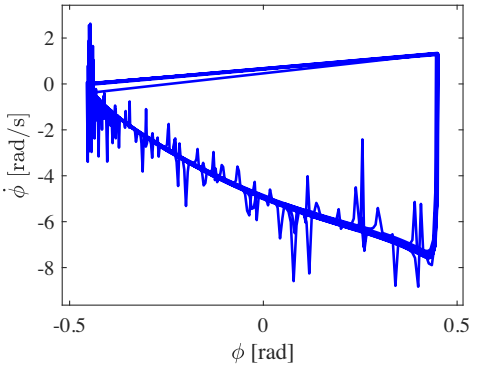



***Figure 43.*** *Phase diagram of $\phi$ and $\dot{\phi}$ for OZC-based steady-state walking in disturbance environment.*

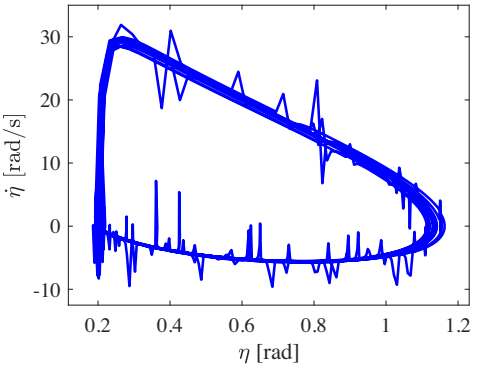



***Figure 44.*** *Phase diagram of $\eta$ and $\dot{\eta}$ for OZC-based steady-state walking in disturbance environment.*

### *4.4. Overall comparison*

To measure the computation time of NMPC and OZC, the original MATLAB programs were converted to C using MEX files, and the computation time per input was evaluated. The average computation times were approximately 0.32 ms for NMPC and 0.026 ms for OZC. Although the computation

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

***Table III.*** *Walking performance.*

| Walk type | Speed (m/s) | SR | Calc. time (ms) |
|---|---|---|---|
| OZC | 0.829 | 1.442 | 0.026 |
| OZC with disturbance | 0.829 | 1.979 | 0.026 |
| NMPC | 0.832 | 1.177 | 0.32 |
| NMPC with disturbance | 0.834 | 1.287 | 0.32 |

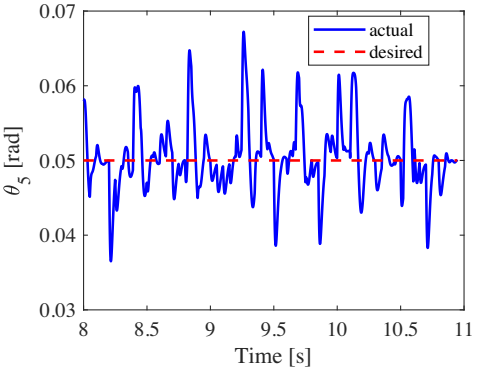



***Figure 45.*** *Variations in the torso angle during NMPC-based walking in disturbance environment with respect to time.*

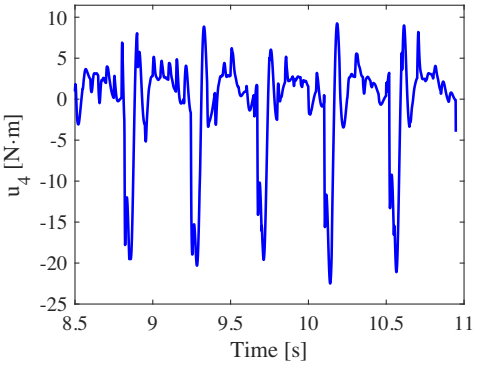



***Figure 46.*** *Variations in $u_4$ during NMPC-based walking in disturbance environment with respect to time.*

time for NMPC is longer than that of OZC, it remains practical even for fast robot control cycles (approximately 1 ms). The results are summarized in Table III. Therefore, the proposed NMPC-based method achieved a walking speed and convergence to the desired trajectory nearly identical to that of the OZC-based approach, with a significant improvement in energy efficiency, particularly in the disturbed environment.

Furthermore, Figures 57 and 58 show the walking speeds of OZC and NMPC-based walking under conditions without and with disturbances when the desired torso angle is varied. Figures 59 and 60 present the SR of OZC and NMPC-based walking with respect to the desired torso angle in these conditions. Additionally, Figures 61 and 62 show the SR of OZC and NMPC-based walking with respect

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

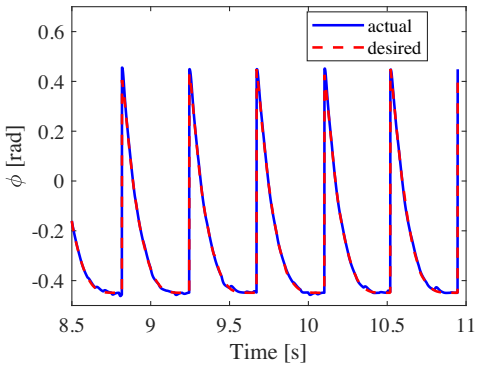



***Figure 47.*** *Variations in $\phi$ during NMPC-based walking in disturbance environment with respect to time.*

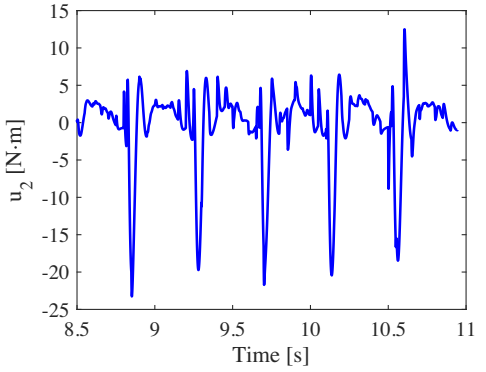



***Figure 48.*** *Variations in $u_2$ during NMPC-based walking in disturbance environment with respect to time.*

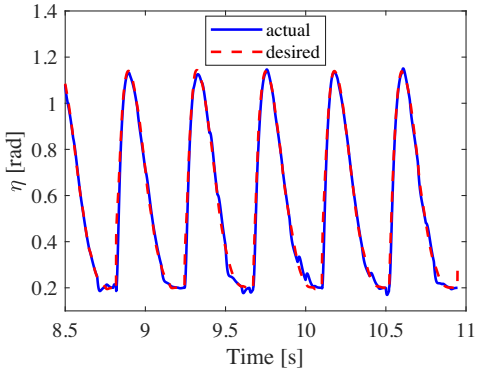



***Figure 49.*** *Variations in $\eta$ during NMPC-based walking in disturbance environment with respect to time.*

to the walking speed in these conditions. It can be observed that the walking speeds of OZC and NMPC remain similar at the same desired torso angle, regardless of the presence of disturbances. However, the SR values of NMPC are consistently lower than those of OZC for all desired torso angles (or the

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

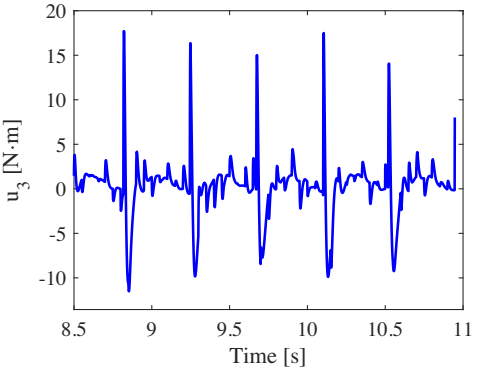



***Figure 50.*** *Variations in $u_3$ during NMPC-based walking in disturbance environment with respect to time.*

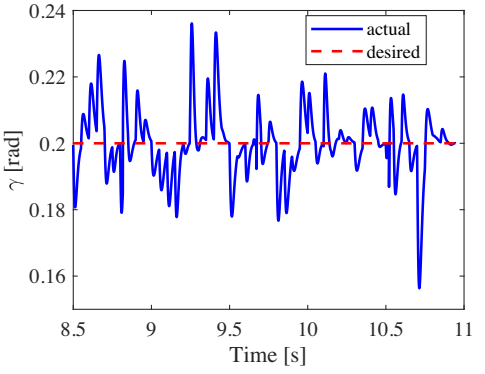



***Figure 51.*** *Variations in $\gamma$ during NMPC-based walking in disturbance environment with respect to time.*

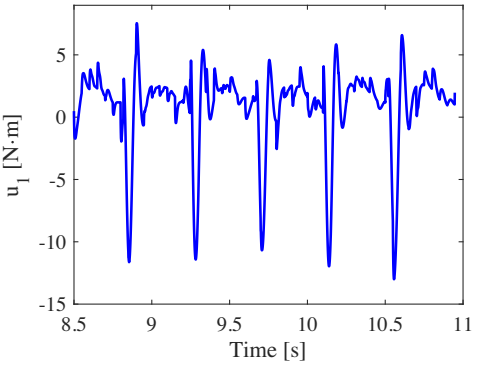



***Figure 52.*** *Variations in $u_1$ during NMPC-based walking in disturbance environment with respect to time.*

walking speed according to the torso angles), with this trend becoming more pronounced in the presence of disturbances. Therefore, the proposed method achieves energy-efficient walking not only locally but also in terms of global walking speed.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

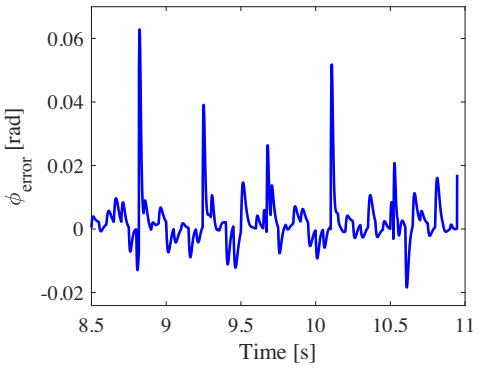



***Figure 53.*** *Variations in the error between the desired value ($\phi_d$) and the actual value during NMPC-based walking in disturbance environment with respect to time.*

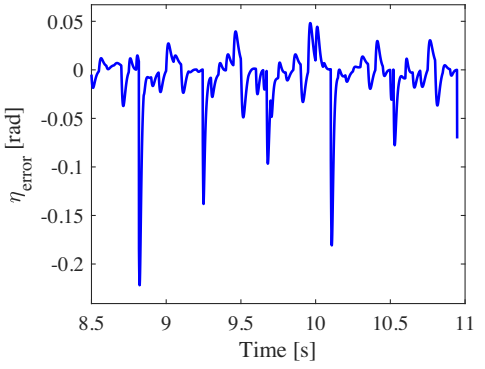



***Figure 54.*** *Variations in the error between the target value ($\eta_d$) and the actual value during NMPC-based walking in disturbance environment with respect to time.*

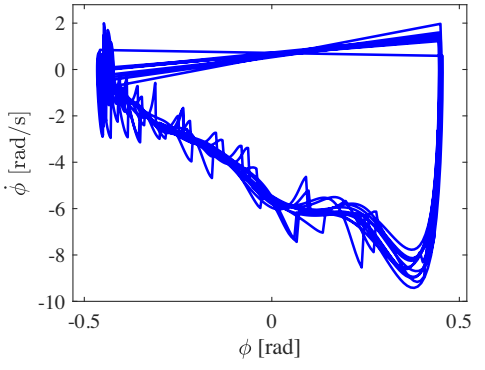



***Figure 55.*** *Phase diagram of $\phi$ and $\dot{\phi}$ for NMPC-based steady-state walking in disturbance environment.*

## 5. Conclusion and future works

This paper presented an NMPC-based method for generating limit cycles suitable for bipedal walking, and the mechanism for generating limit cycles was explained in detail. Specifically, it uses the proposed NMPC to attract the robot's state to a virtual constraint trajectory based on the support leg angle.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

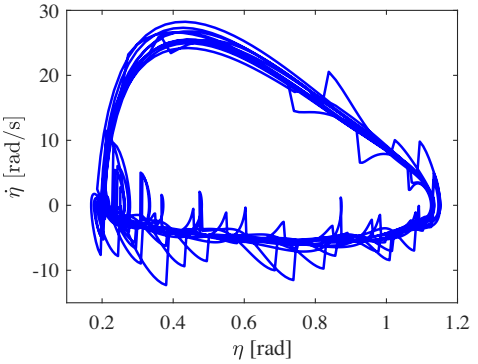



***Figure 56.*** *Phase diagram of $\eta$ and $\dot{\eta}$ for NMPC-based steady-state walking in disturbance environment.*

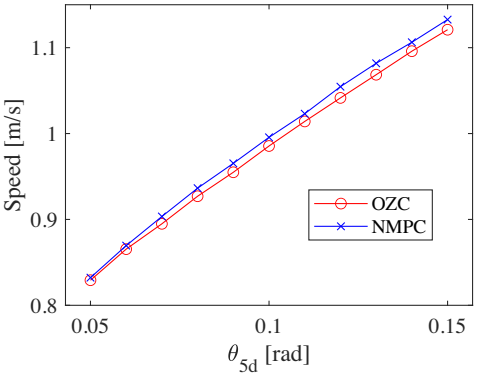



***Figure 57.*** *Walking speeds for OZC and NMPC steady-state walking with respect to $\theta_{5d}$.*

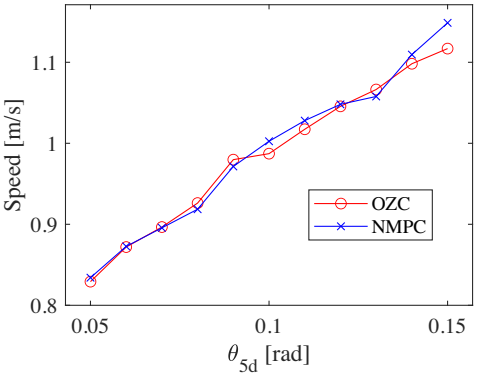



***Figure 58.*** *Walking speeds for OZC and NMPC steady-state walking in disturbance environment with respect to $\theta_{5d}$.*

To demonstrate the effectiveness of the proposed method, numerical simulations were conducted to compare it with the conventional OZC method. The results showed that the proposed method not only exhibits convergence to the desired trajectory similar to the conventional method but also offers significant advantages in terms of energy efficiency. In particular, our approach demonstrated substantial

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

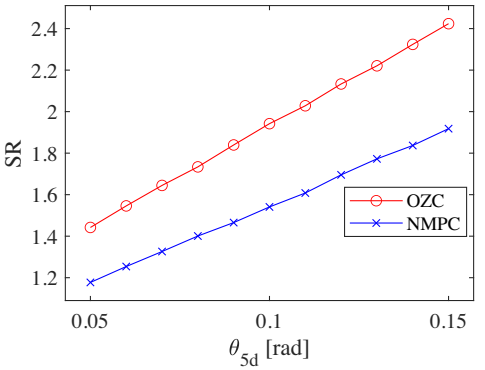



***Figure 59.*** *SR values for OZC and NMPC steady-state walking with respect to* $\theta_{5d}$.

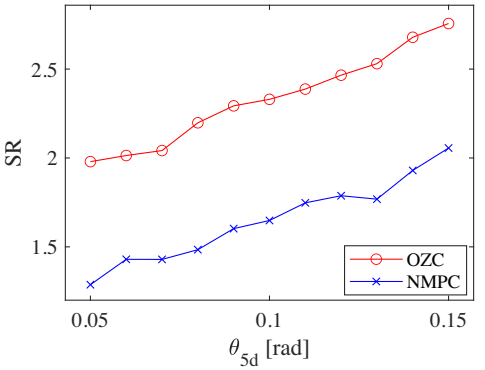



***Figure 60.*** *SR values for OZC and NMPC steady-state walking in disturbance environment with respect to* $\theta_{5d}$.

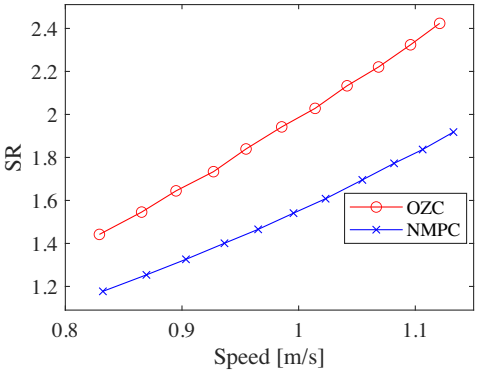



***Figure 61.*** *SR values for OZC and NMPC steady-state walking with respect to walking speed.*

improvements in environments with disturbances, successfully mitigating their impact on energy efficiency. Additionally, the computation time required by the proposed method was analyzed, and it was confirmed that the calculations can be performed within a practical control cycle. Although this study did not explore trajectory optimization, the convergence to the target trajectory was comparable to that of

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

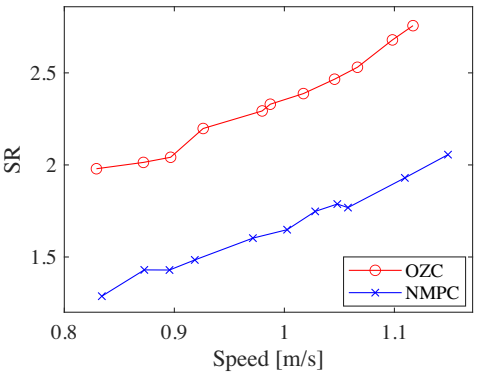



***Figure 62.*** *SR values for OZC and NMPC steady-state walking in disturbance environment with respect to walking speed.*

high-gain OZC, suggesting that HZD-based optimization is feasible. Therefore, future work will focus on trajectory optimization using HZD and extension of the method to three-dimensional walking by incorporating coronal plane motions.

**Supplementary material.** The supplementary material for this article can be found at https://doi.org/10.1017/S0263574725103020.

**Author contributions.** Y. Hanazawa conceived and designed this study. Y. Hanazawa and Y. Uchino verified the effectiveness of the proposed method. Y. Hanazawa wrote the article. S. Sagara provided technical support for the study.

**Financial support.** Not support.

**Competing interest.** The authors declare no conflicts of interest.

**Use of AI-assisted tools.** The authors used Claude Sonnet 4.5 only for English language editing to improve readability.

**Ethical approval.** Not applicable.

## References

[1] L. Krishna, G. A. Castillo, U. A. Mishra, A. Hereid and S. Kolathaya, "Linear policies are sufficient to realize robust bipedal walking on challenging terrains," *IEEE Rob. Autom. Lett.* **7**(2), 2047–2054 (2022).

[2] E. Selim, M. Alcı and M. Altıntas, "Variable-time-interval trajectory optimization-based dynamic walking control of bipedal robot," *Robotica* **40**(6), 1799–1819 (2022).

[3] M. S. Khan and R. K. Mandava, "A review on gait generation of the biped robot on various terrains," *Robotica* **41**(6), 1888–1930 (2023).

[4] L. Han, X. Chen, Z. Yu, Z. Gao, G. Huang, J. Zhang, K. Hashimoto and Q. Huang, "A heuristic gait template planning and dynamic motion control for biped robots," *Robotica* **41**(2), 789–805 (2023).

[5] Y. Xie, C. Gao, S. Zhu, X. Yan, L. Kong, A. Xie, J. Gu and D. Zhang, "Gait optimization and energy-based stability for biped locomotion using large-scale programming," *Robotica* **41**(7), 2207–2227 (2023).

[6] N. Csomay-Shanklin, W. D. Compton, I. D. J. Rodriguez, E. R. Ambrose, Y. Yue and A. D. Ames, "Robust Agility via Learned Zero Dynamics Policies," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2024) pp. 9116–9123.

[7] H. Y. Park and J. H. Kim, "Robust Balancing Control of Biped Robots for External Forces," *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2024) pp. 13257–13262.

[8] A. Dan, S. K. Saha and K. R. Krishna, "Stable underactuated biped locomotion on various geometrical surfaces," *Multibody Syst. Dyn.* **63**(1), 183–208 (2025).

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, "Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. **2** (2003) pp. 1620–1626.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

[10] K. Nishiwaki and S. Kagami, "Strategies for Aadjusting the ZMP Reference Trajectory for Maintaining Balance in Humanoid Walking," *IEEE International Conference on Robotics and Automation (ICRA)* (2010) pp. 4230–4236.
[11] S. Shigemi, A. Goswami and P. Vadakkepat, "Asimo and Humanoid Robot Research at Honda," **In:** *Humanoid Robotics: A Reference* (2018) pp. 55–90.
[12] D. Rodriguez and S. Behnke, "Deepwalk: Omnidirectional Bipedal Gait by Deep Reinforcement Learning," *2021 IEEE international conference on robotics and automation (ICRA)* (2021) pp. 3033–3039. IEEE.
[13] M. Kasaei, M. Abreu, N. Lau, A. Pereira and L. P. Reis, "Robust biped locomotion using deep reinforcement learning on top of an analytical control approach," *Rob. Auton. Syst.* **146**, 103900 (2021).
[14] H. Ahmad, Y. Nakata, Y. Nakamura and H. Ishiguro, "Adjustable whole-body dynamics for adaptive locomotion: The influence of upper body movements and its interactions with the lower body parts on the stable locomotion of a simple bipedal robot," *Robotica* **40**(9), 3340–3354 (2022).
[15] S. E. Ada, E. Ugur and H. L. Akin, "Generalization in transfer learning: Robust control of robot locomotion," *Robotica* **40**(11), 3811–3836 (2022).
[16] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *Int. J. Rob. Res.* **44**(5), 840–888 (2025).
[17] J. Pratt, J. Carff, S. Drakunov and A. Goswami, "Capture Point: A Step Toward Humanoid Push Recovery," *2006 6th IEEE-RAS International Conference on Humanoid Robots* (IEEE , 2006) pp. 200–207.
[18] T. Koolen, T. De Boer, J. Rebula, A. Goswami and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *Int. J. Rob. Res.* **31**(9), 1094–1113 (2012).
[19] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Auton. Rob.* **40**(3), 473–491 (2016).
[20] S. Collins, A. Ruina, R. Tedrake and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science* **307**(5712), 1082–1085 (2005).
[21] D. G. Hobbelen and M. Wisse, Limit cycle walking (2007).
[22] E. R. Westervelt, J. W. Grizzle and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Trans. Autom. Control* **48**(1), 42–56 (2003).
[23] J. W. Grizzle, G. Abba and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE Trans. Autom. Control* **46**(1), 51–64 (2001).
[24] J. W. Grizzle, J. Hurst, B. Morris, H.-W. Park and K. Sreenath, "Mabel, A New Robotic Bipedal Walker and Runner," *2009 American Control Conference* (IEEE, 2009) pp. 2030–2036.
[25] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang and J. Grizzle, "Feedback Control of a Cassie Bipedal Robot: Walking, Standing, and Riding a Segway," *American Control Conference (ACC)* (2019) pp. 4559–4566.
[26] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *J. Dyn. Syst. Meas. Control* **144**(12), 121006 (2022).
[27] T. Shimizu, S. Nakaura and M. Sampei, "The Control of a Bipedal Running Robot Based on Output Zeroing Considered Rotation of the Ankle Joint," *Proceedings of the 45th IEEE Conference on Decision and Control (CDC)* (2006) pp. 6456–6461.
[28] K. Nakamura, S. Nakaura and M. Sampei, "Control of Bipedal Running by the Angular-Momentum-based Synchronization Structure," *IEEE International Conference on Robotics and Automation (ICRA)* (2010) pp. 3310–3315.
[29] A. Hereid, E. A. Cousineau, C. M. Hubicki and A. D. Ames, "3D Dynamic Walking with Underactuated Humanoid Robots: A Direct Collocation Framework for Optimizing Hybrid Zero Dynamics," *IEEE International Conference on Robotics and Automation (ICRA)* (2016) pp. 1447–1454.
[30] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *Int. J. Rob. Res.* **35**(12), 1497–1521 (2016).
[31] R. Takano, J. Chang and M. Yamakita, "Periodic Trajectory Planning and Robust Output Zeroing Control for Underactuated Bipedal Robots with Predicted Disturbances," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019) pp. 1415–1420.
[32] J. Reher and A. D. Ames, "Inverse Dynamics Control of Compliant Hybrid Zero Dynamic Walking," *2021 IEEE International Conference on Robotics and Automation (ICRA)* ( 2021) pp. 2040–2047.
[33] Y. Luo, U. J. Römer, A. Dyck, M. Zirkel, L. Zentner and A. Fidlin, "Hybrid zero dynamics control for bipedal walking with a non-instantaneous double support phase," arXiv preprint arXiv: 2303.05165 (2023).
[34] A. B. Ghansah, J. Kim, K. Li and A. D. Ames, "Dynamic Walking on Highly Underactuated Point Foot Humanoids: Closing the Loop Between HZD and HLIP," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2024) pp. 12686–12693.
[35] T. McGeer, "Passive dynamic walking," *Int. J. Rob. Res.* **9**(2), 62–82 (1990).
[36] Y. Hanazawa, H. Nishinami and S. Sagara, "Energy-efficient limit cycle walking in disturbance based on nonlinear model predictive control," *Robotica* **42**(2), 435–456 (2024).
[37] T. Ohtsuka, "A continuation/gmres method for fast computation of nonlinear receding horizon control," *Automatica* **40**(4), 563–574 (2004).
[38] H. Deng and T. Ohtsuka, "A parallel newton-type method for nonlinear model predictive control," *Automatica* **109**, 1–11, 108560 (2019).
[39] Y. Onodera and M. Yamakita, "An Extension of Nonlinear Receding Horizon Control for Switched System with State Jump," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2005) pp. 984–989.

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

[40] A. Murayama and M. Yamakita, "Speed Control of Vehicles with Variable Valve Lift Engine by Nonlinear MPC," *IEEE ICCAS-SICE* (2009) pp. 4128–4133.
[41] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi and B. Morris, "Feedback control of dynamic bipedal robot locomotion" (2018).
[42] P. Lu, "Optimal predictive control of continuous nonlinear systems," *Int. J. Control* **62**(3), 633–649 (1995).

## Appendix

### *Details of the model*

The components of matrices $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$, and $\boldsymbol{G}(\boldsymbol{q})$ are as follows, where $\boldsymbol{H}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q})$, $M_{cij}$ denotes the $(i,j)$-th component of $\boldsymbol{M}(\boldsymbol{q})$, $C_{ci}$ denotes the $i$-th row component of $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$, and $G_{ci}$ denotes the $i$-th row component of $\boldsymbol{G}(\boldsymbol{q})$.

$$
\begin{aligned}
M_{c11} &= I_1 + a_1^2 m_1 + l_1^2 m_1 + 2l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_h \\
M_{c12} &= l_1 \cos(\theta_1 - \theta_2)(a_2 m_2 + l_2 m_1 + l_2 m_2 + l_2 m_3 + l_2 m_h) \\
M_{c13} &= -l_1 \cos(\theta_1 - \theta_3)(b_2 m_2 + l_2 m_1) \\
M_{c14} &= -b_1 l_1 m_1 \cos(\theta_1 - \theta_4) \\
M_{c15} &= a_3 l_1 m_3 \cos(\theta_1 - \theta_5) \\
M_{c21} &= l_1 \cos(\theta_1 - \theta_2)(a_2 m_2 + l_2 m_1 + l_2 m_2 + l_2 m_3 + l_2 m_h) \\
M_{c22} &= I_2 + a_2^2 m_2 + l_2^2 m_1 + l_2^2 m_2 + l_2^2 m_3 + l_2^2 m_h \\
M_{c23} &= -l_2 \cos(\theta_2 - \theta_3)(b_2 m_2 + l_2 m_1) \\
M_{c24} &= -b_1 l_2 m_1 \cos(\theta_2 - \theta_4) \\
M_{c25} &= a_3 l_2 m_3 \cos(\theta_2 - \theta_5) \\
M_{c31} &= -l_1 \cos(\theta_1 - \theta_3)(b_2 m_2 + l_2 m_1) \\
M_{c32} &= -l_2 \cos(\theta_2 - \theta_3)(b_2 m_2 + l_2 m_1) \\
M_{c33} &= I_2 + b_2^2 m_2 + l_2^2 m_1 \\
M_{c34} &= b_1 l_2 m_1 \cos(\theta_3 - \theta_4) \\
M_{c35} &= 0 \\
M_{c41} &= -b_1 l_1 m_1 \cos(\theta_1 - \theta_4) \\
M_{c42} &= -b_1 l_2 m_1 \cos(\theta_2 - \theta_4) \\
M_{c43} &= b_1 l_2 m_1 \cos(\theta_3 - \theta_4) \\
M_{c44} &= I_1 + b_1^2 m_1 \\
M_{c45} &= 0 \\
M_{c51} &= a_3 l_1 m_3 \cos(\theta_1 - \theta_5) \\
M_{c52} &= a_3 l_2 m_3 \cos(\theta_2 - \theta_5) \\
M_{c53} &= 0 \\
M_{c54} &= 0 \\
M_{c55} &= I_3 + a_3^2 m_3 \\
C_{c1} &= l_1(a_2 \dot{\theta}_2^2 m_2 \sin(\theta_1 - \theta_2) + a_3 \dot{\theta}_5^2 m_3 \sin(\theta_1 - \theta_5) - b_1 \dot{\theta}_4^2 m_1 \sin(\theta_1 - \theta_4) - b_2 \dot{\theta}_3^2 m_2 \sin(\theta_1 - \theta_3) \\
&\quad + \dot{\theta}_2^2 l_2 m_1 \sin(\theta_1 - \theta_2) + \dot{\theta}_2^2 l_2 m_2 \sin(\theta_1 - \theta_2) + \dot{\theta}_2^2 l_2 m_3 \sin(\theta_1 - \theta_2) \\
&\quad - \dot{\theta}_3^2 l_2 m_1 \sin(\theta_1 - \theta_3) + \dot{\theta}_2^2 l_2 m_h \sin(\theta_1 - \theta_2)) \\
C_{c2} &= a_3 \dot{\theta}_5^2 l_2 m_3 \sin(\theta_2 - \theta_5) - a_2 \dot{\theta}_1^2 l_1 m_2 \sin(\theta_1 - \theta_2) - \dot{\theta}_3^2 l_2^2 m_1 \sin(\theta_2 - \theta_3) - b_1 \dot{\theta}_4^2 l_2 m_1 \sin(\theta_2 - \theta_4) \\
&\quad - b_2 \dot{\theta}_3^2 l_2 m_2 \sin(\theta_2 - \theta_3) - \dot{\theta}_1^2 l_1 l_2 m_1 \sin(\theta_1 - \theta_2) - \dot{\theta}_1^2 l_1 l_2 m_2 \sin(\theta_1 - \theta_2) \\
&\quad - \dot{\theta}_1^2 l_1 l_2 m_3 \sin(\theta_1 - \theta_2) - \dot{\theta}_1^2 l_1 l_2 m_h \sin(\theta_1 - \theta_2)
\end{aligned}
$$

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press

$$
\begin{aligned}
C_{c3} &= \dot{\theta}_2^2 l_2^2 m_1 \sin(\theta_2 - \theta_3) + b_2 \dot{\theta}_1^2 l_1 m_2 \sin(\theta_1 - \theta_3) + b_2 \dot{\theta}_2^2 l_2 m_2 \sin(\theta_2 - \theta_3) \\
&\quad + b_1 \dot{\theta}_4^2 l_2 m_1 \sin(\theta_3 - \theta_4) + \dot{\theta}_1^2 l_1 l_2 m_1 \sin(\theta_1 - \theta_3) \\
C_{c4} &= b_1 m_1 (\dot{\theta}_1^2 l_1 \sin(\theta_1 - \theta_4) + \dot{\theta}_2^2 l_2 \sin(\theta_2 - \theta_4) - \dot{\theta}_3^2 l_2 \sin(\theta_3 - \theta_4)) \\
C_{c5} &= -a_3 m_3 (\dot{\theta}_1^2 l_1 \sin(\theta_1 - \theta_5) + \dot{\theta}_2^2 l_2 \sin(\theta_2 - \theta_5)) \\
G_{c1} &= -g \sin\theta_1 (a_1 m_1 + l_1 m_1 + 2 l_1 m_2 + l_1 m_3 + l_1 m_h) \\
G_{c2} &= -g \sin\theta_2 (a_2 m_2 + l_2 m_1 + l_2 m_2 + l_2 m_3 + l_2 m_h) \\
G_{c3} &= g \sin\theta_3 (b_2 m_2 + l_2 m_1) \\
G_{c4} &= b_1 m_1 \sin\theta_4 g \\
G_{c5} &= -a_3 m_3 \sin\theta_5 g
\end{aligned}
$$

**Cite this article:** Y. Hanazawa, Y. Uchino and S. Sagara (2026). "Limit cycles generation for energy-efficient bipedal walking via nonlinear model predictive control", Robotica **44**, 1–36. https://doi.org/10.1017/S0263574725103020

https://doi.org/10.1017/S0263574725103020 Published online by Cambridge University Press