Robotica (2025), **43**, pp. 1258–1290
doi:10.1017/S0263574725000244

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Adaptive optimization for virtual model control of quadruped robots based on BP neural network

Jianwen Liu, Xiaojun Xu, Wenhao Wang, Yuanjiang Tang and Shengyang Lu

College of Intelligent Science, National University of Defense Technology, Changsha, PR China
Corresponding author: Jianwen Liu; Email: liujianwen17@nudt.edu.cn

Received: 12 September 2024; **Revised:** 31 December 2024; **Accepted:** 29 January 2025;
First published online: 26 February 2025

Keywords: quadruped robot; virtual model control; BP neural network; parameter adaptation; control of robotic systems

Abstract

The virtual model control (VMC) method establishes a direct correlation model between the end-effector and the main body by selecting appropriate virtual mechanical components. This approach facilitates direct force control while circumventing the necessity for complex dynamic modeling. However, the simplification inherent in this modeling can result in inaccuracies in the calculation of joint driving torques, ultimately diminishing control precision. Moreover, VMC typically depends on predefined models for control, which constrains its adaptability in dynamically complex environments and under varying movement conditions. To address these limitations, this paper proposes the BP-VMC method, which is based on a backpropagation neural network (BPNN). Initially, a quadruped robot model was established through kinematic analysis. Subsequently, a decomposed VMC model was developed, and BPNN was introduced to facilitate the adaptive tuning of virtual parameters. This approach resulted in the creation of a virtual mechanical component model with adaptive capabilities, compensating for errors arising from simplified modeling. Finally, a simulation control system was constructed based on the BP-VMC control framework to validate the optimization of control performance. Simulation experiments demonstrated that, in comparison to traditional VMC methods, the BP-VMC method exhibits enhanced control accuracy and stability, achieving a 30% reduction in trajectory tracking error, a 40% reduction in velocity tracking error, and a 20–30% improvement in instability indices across various working conditions. This evidence underscores the BP-VMC method's robust adaptability in dynamic environments.

1. Introduction

Research on legged robots has its origins in the 1960s. In comparison to wheeled and tracked robots, legged robots demonstrate superior adaptability to various terrains and enhanced capabilities for traversing obstacles, which allows them to perform tasks in intricate environments [1–4]. As advancements in robot control technology continue, legged robots have increasingly emerged as a focal point of research interest [5–7]. These robots are characterized as time-varying, strongly coupled, and multivariable dynamic systems that are significantly affected by gravitational forces, resulting in substantial computational requirements for motion control. Among the different types of legged robots, quadruped robots are noted for their exceptional performance in terms of load-bearing capacity, diverse movement patterns, and effective motion control [8]. Presently, the control strategies employed for quadruped robots can be broadly categorized into two main types: model-based control methods and model-independent control methods [9, 10].

Model-based control methodologies generally entail the development of a model of the robotic system through the analysis of kinematics and dynamics, followed by the planning of a desired trajectory [11, 12]. Subsequently, feedback control is employed to progressively align the actual motion trajectory with the intended trajectory. Notable examples of model-based control techniques include Zero Moment Point control [13, 14], Three-Part Control (3 PC) [15, 16], and virtual model control (VMC) [17, 18].

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

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

In contrast, model-independent control strategies do not depend on the dynamic model of the robot; rather, they operate based on input-output signals and pertinent real-time data. This characteristic allows for the maintenance of stable robot operation even in the presence of model inaccuracies or uncertainties. A prominent example of this approach is the central pattern generator-based motion control method [19–22]. With advancements in machine learning technologies, an increasing number of researchers have introduced model-independent control strategies that leverage intelligent algorithms [23–26].

An examination of pertinent research findings reveals that model-independent control methods operate independently of precise dynamic models. These methods attain control objectives by continuously modifying the control output in response to feedback derived from interactions with the environment, demonstrating significant adaptability. However, the absence of guidance from specific models poses challenges in maintaining reliable control stability within complex environments. Conversely, model-based control methods depend on dynamic models, which facilitate robots in executing desired motion trajectories with enhanced accuracy, control precision, and stability. Theoretically, optimal control outcomes can be realized when based on an accurate dynamic model. Nevertheless, in practical applications, the dynamic factors involved are often intricate and complex, complicating the modeling process. Furthermore, the models that are developed typically pertain to specific operational conditions, thereby constraining their adaptability in multifaceted environments.

Taking all factors into consideration, the VMC method belongs to the category of model-based control methods, but it simplifies the dynamic modeling process by directly establishing a correlation model between the end-effector and the robot body. Furthermore, VMC constructs virtual models in the form of equivalent and universal mechanical components, which possess broad generality. By adjusting relevant virtual parameters, it can be adapted to different motion control models. Therefore, it is feasible to propose an optimized adaptive VMC method that realizes model-based control while enhancing adaptability and robustness.

The VMC method approaches from the perspective of intuitive control, envisioning virtual mechanical components connecting the robot's body (internal action points) with the external environment (external action points), such that the generated virtual forces can drive the robot to perform corresponding actions. Then, based on the principle of virtual work, the virtual forces are mapped into torques in the joint space, serving as the output targets for the joint motors. Chew [27] implemented walking control for biped robots based on VMC and proposed an adaptive VMC using sliding mode control theory. Later, Massachusetts Institute Of Technology, combining model predictive technology, achieved control of complex actions like jumping on the Cheetah3 robot [28, 29]. Zhang [30] et al. proposed a joint torque servo and active compliance control method for robot legs based on VMC, and subsequently, Chen [31] et al. transformed the force distribution problem into a QP optimization problem to achieve motion control under optimal foot force.

Numerous research initiatives pertaining to VMC concentrate on the prediction and allocation of the requisite virtual forces, thereby optimizing control during the final output phase. Nonetheless, the formulation of the virtual model, particularly the selection of virtual parameters, exerts a direct influence on the control efficacy of VMC. Variations in virtual models yield distinct dynamic characteristics within VMC systems. The attainment of adaptive modeling of the virtual model, tailored to the specific demands of motion and the fluctuations in the motion environment, can substantially improve the adaptability of the VMC controller and enhance its overall control performance [32, 33].

Therefore, this paper proposes a VMC method based on backpropagation neural network (BPNN) optimization from the perspective of constructing virtual models. In this method, a BPNN is used to equivalently replace the virtual parameter matrix in VMC. Leveraging the self-training capability of BPNN, adaptive tuning of virtual parameters, that is, adaptive modeling of virtual mechanical components, is achieved. Depending on changes in the environment or motion state, BPNN can also effectively achieve adaptive optimization of the model and virtual parameters, thus improving the adaptability of the controller. The main achievements are:

1. Adaptive tuning of virtual parameters, which realizes adaptive modeling of the virtual component model. This enables the VMC controller to automatically adjust parameters based on different environments and task requirements, improving the robot's adaptability and flexibility.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

2. Optimized trajectory and velocity tracking performance, effectively reducing tracking errors. This ensures that the robot can maintain precise and stable movement under different motion targets.
3. Enhanced control stability and motion stability. The robot can maintain a stable operating state even in dynamic environments, reducing the risk of losing control.

The framework of this article is shown in Figure 1. In Sections 2 and 3, a quadruped robot model is established based on kinematic analysis, and a virtual control model is constructed accordingly. In Sections 4 and 5, the control systems and frameworks for VMC and BP-VMC are developed. In Section 6, the improved BP-VMC control algorithm is applied to the gait planning and motion trajectory control of the quadruped robot, and a comparative analysis is conducted with VMC as the control group.

2. Quadruped robot model

In the control of leg movement, the more degrees of freedom (DOF) a leg has, the more flexible the movement becomes, but also the more complex the control becomes. Considering the primary movement needs of quadruped robots, each leg typically has three DOFs, which, respectively, achieve hip abduction, thigh swing, and shank swing. The hip abduction DOF mainly serves as a corrective control function based on stability. Therefore, when analyzing the kinematic model of a quadruped robot during locomotion, the hip abduction DOF is temporarily excluded. The main focus is on analyzing the thigh and shank swing DOFs.

The quadruped robot model studied in this paper adopts a front elbow-rear knee configuration, as shown in Figure 2. Compared to other leg configurations, this configuration exhibits a more compact overall structure, a uniform distribution of mass, and reduced self-motion fluctuations, all of which contribute to the stability of the robot during its movement. To facilitate the subsequent description of the positional relationships between various joint points, three coordinate systems are defined: the body coordinate system $B-XYZ$ with the centroid of the body as the reference, the hip coordinate system $H-xyz$ with the hip joint as the origin and the y-axis aligned with the rotation axis, and the foot-end coordinate $A-xyz$ system with the end-point A of the leg as the origin.

In the process of walking motion control, the main approach is to control the rotation angles of the thigh and shank swing to achieve changes in the position output of the leg's end-effector, thus fulfilling the purpose of controlling the legged movement of the robot. This involves issues related to the kinematics analysis of the robot, which requires both the ability to derive the end-effector position output based on the input of the actuated joints and to determine the required input of the actuated joints based on the desired position output of the end-effector. This ensures the certainty of the robot's motion control.

Due to the symmetrical structure of the legs of the quadruped robot, the right rear leg is selected for analysis without loss of generality. The single-leg structure is shown in Figure 3. θ_0 represents the angle of the passive joint established at the interface between the foot-end and the ground, θ_1 and θ_2 represent the angles of the active joints, A is the endpoint of the leg, H is the hip joint of the robot's leg, B is the center of mass of the robot, and the subscript $i(i=1,2,3,4)$ indicates relevant parameters of the ith leg.

Let the lengths of the four rods be represented by the variables l_1, l_2, l_3 and l_4, with the stipulation that $l_1=l_3$. The length of the robot body is denoted as $2l_0$, the height of the centroid is denoted as h_0, the length of the thigh is denoted as l_a, and the length of the shank is denoted as l_b. The relationships among these variables are articulated through the equations $l_a=l_1$ and $l_b=l_4-l_2$. The forward kinematics mapping relationship from point A to point B can be derived as follows:

$$
{}^{A}_{B}T=\begin{bmatrix} x \\ z \\ \theta \end{bmatrix}=\begin{bmatrix} l_b c\theta_0 - l_a c\left(\theta_2-\theta_1-\theta_0\right)+l_0 c\left(\theta_0-\theta_2\right)-h_0 s\left(\theta_0-\theta_2\right) \\ l_b s\theta_0 + l_a s\left(\theta_2-\theta_1-\theta_0\right)+l_0 s\left(\theta_0-\theta_2\right)+h_0 c\left(\theta_0-\theta_2\right) \\ \theta_0-\theta_2 \end{bmatrix} \tag{1}
$$

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

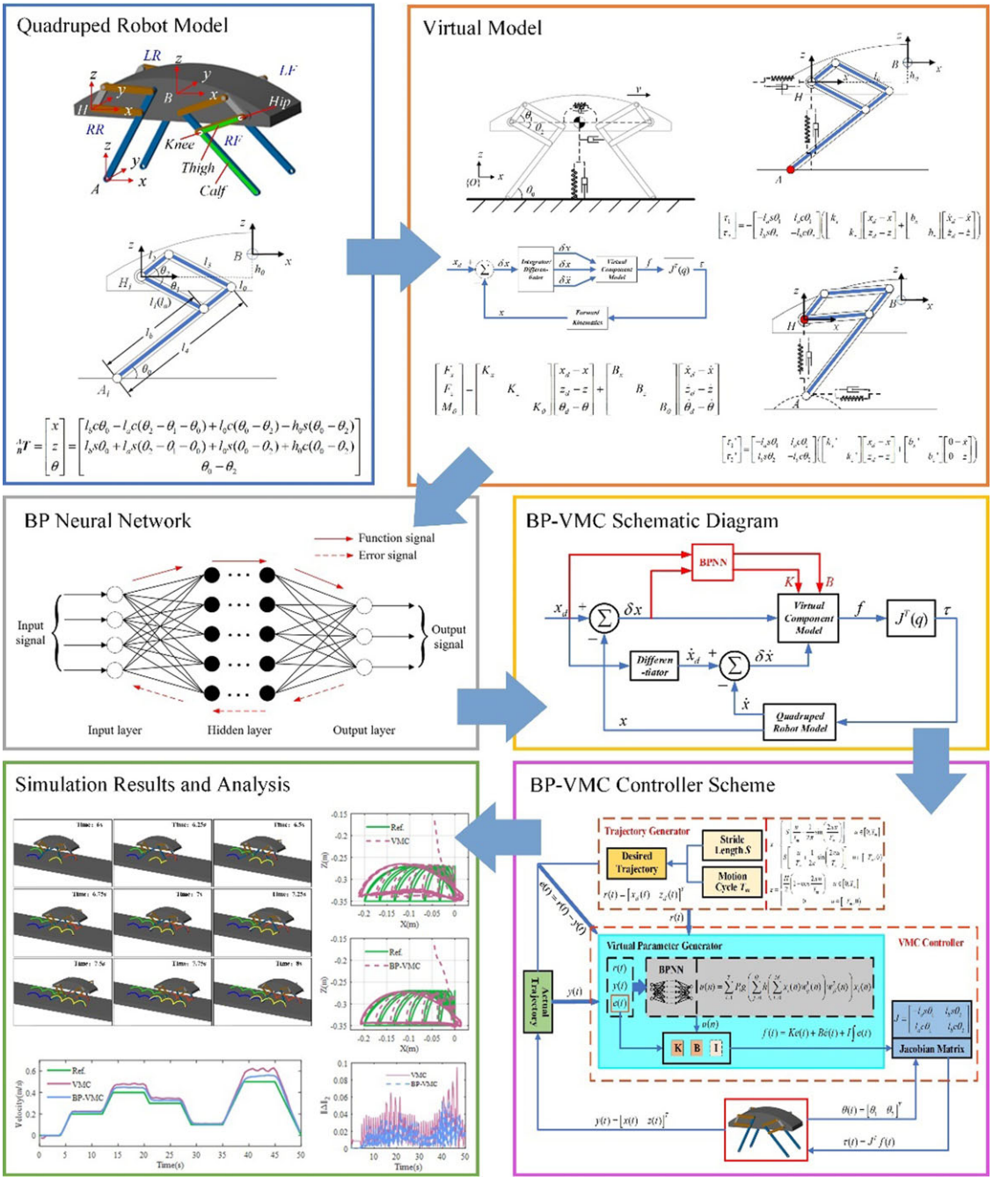


Figure 1. *Framework of this paper.*

By taking the partial derivatives of the joint angles, the Jacobian matrix is obtained as follows:

$$
{}^{A}_{B}J = \begin{bmatrix} -l_b s\theta_0 - l_a s\theta_{+2-1-0} - l_0 s\theta_{+0-2} - h_0 c\theta_{+0-2} & -l_a s\theta_{+2-1-0} & l_a s\theta_{+2-1-0} + l_0 s\theta_{+0-2} + h_0 c\theta_{+0-2} \\ l_b c\theta_0 - l_a c\theta_{+2-1-0} + l_0 c\theta_{+0-2} - h_0 s\theta_{+0-2} & -l_a c\theta_{+2-1-0} & l_a c\theta_{+2-1-0} - l_0 c\theta_{+0-2} + h_0 s\theta_{+0-2} \\ 1 & 0 & -1 \end{bmatrix} \tag{2}
$$

where $\theta_{+2-1-0} = \theta_2 - \theta_1 - \theta_0$, and the same applies to other instances.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

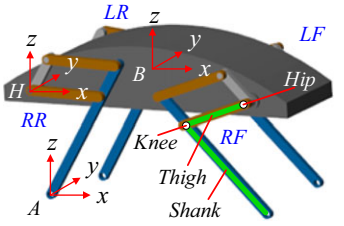


Figure 2. *Quadruped robot model.*

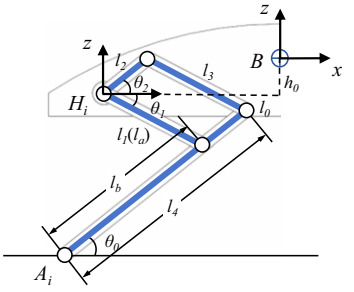


Figure 3. *Analysis of single-leg structure.*

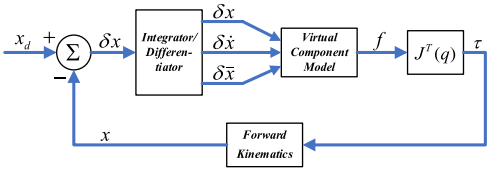


Figure 4. *VMC algorithm framework.*

In practical applications, the establishment of an accurate dynamics model is challenging due to the frequent fluctuations in the robot's motion state and the presence of various external environmental disturbances. As shown in Figure 4, the VMC algorithm ignores the dynamic factors of the robot and constructs a virtual component model. It establishes a correlation model between the control force exerted by the end-effector and the robot's body utilizing the Jacobian matrix, thereby illustrating superior characteristics in force control. This method fully considers the generalized coordinates of the robot's body and end-effector, effectively achieving control over the robot's body motion state.

3. Virtual model control

3.1 VMC model

The core idea of VMC is to analyze the main effective force based on the overall motion control objectives of the robot. This approach disregards the specific driving mechanisms of the robot and instead employs appropriate virtual mechanical components to produce an equivalent force, thereby yielding a virtual force. Utilizing the Jacobian matrix, this virtual force is subsequently transformed into actual driving torques, facilitating the simplification of the robot's motion control model.

The motion control of a robotic system primarily encompasses the regulation of forward velocity along the X direction, the management of the centroid elevation along the Z direction, and the adjustment of the tilt angle about the centroid. Therefore, three sets of virtual component models are constructed accordingly to connect the external action points (environment) with the internal action points (robot body) to achieve control of the three DOFs. The principle structure is shown in Figure 5. The focus in the

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

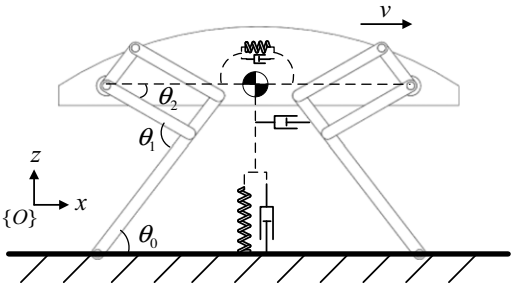


Figure 5. *The VMC principle structure.*

X direction is on velocity control, so only a virtual damping component is applied. However, accurate control of the centroid position is required for the centroid height in the Z direction and the tilt angle of the robot's centroid, so both virtual spring and damping components are applied simultaneously.

Let the desired state of the robot be denoted as $X_d = [\, x_d \;\; z_d \;\; \theta_d \,]^T$ and the current state as$X = [\, x \;\; z \;\; \theta \,]^T$. The virtual forces and torques generated from the transition from the current state to the desired state are

$$\begin{bmatrix} F_x \\ F_z \\ M_\theta \end{bmatrix} = \begin{bmatrix} K_x & & \\ & K_z & \\ & & K_\theta \end{bmatrix} \begin{bmatrix} x_d - x \\ z_d - z \\ \theta_d - \theta \end{bmatrix} + \begin{bmatrix} B_x & & \\ & B_z & \\ & & B_\theta \end{bmatrix} \begin{bmatrix} \dot{x}_d - \dot{x} \\ \dot{z}_d - \dot{z} \\ \dot{\theta}_d - \dot{\theta} \end{bmatrix} \tag{3}$$

where K_x, K_z, and K_θ represent the virtual spring stiffness in the X, Z, and θ directions, respectively, while B_x, B_z, and B_θ represent the virtual damping coefficients in the X, Z, and θ directions, respectively.

In the horizontal direction, only dampers exist, so $K_x = 0$. The directions of these virtual components are all referenced from the ground coordinate system (absolute coordinate system) $\{O\}$.

The desired joint torque is denoted as $\boldsymbol{\tau}_d = [\, \tau_0 \quad \tau_1 \quad \tau_2 \,]^T$. Since θ_0 is a passive joint, $\tau_0 = 0$.

Based on the formula $\boldsymbol{\tau} = J^T F$, it yields

$$\begin{bmatrix} 0 \\ \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} -l_b s\theta_0 - l_a s\theta_{+2-1-0} - l_0 s\theta_{+0-2} - h_0 c\theta_{+0-2} & -l_a s\theta_{+2-1-0} & l_a s\theta_{+2-1-0} + l_0 s\theta_{+0-2} + h_0 c\theta_{+0-2} \\ l_b c\theta_0 - l_a c\theta_{+2-1-0} + l_0 c\theta_{+0-2} - h_0 s\theta_{+0-2} & -l_a c\theta_{+2-1-0} & l_a c\theta_{+2-1-0} - l_0 c\theta_{+0-2} + h_0 s\theta_{+0-2} \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} F_x \\ F_z \\ M_\theta \end{bmatrix} \tag{4}$$

It can be derived from the above formula that

$$F_x = \frac{1}{l_b s\theta_0 + l_a s\theta_{+2-1-0} + l_0 s\theta_{+0-2} + h_0 c\theta_{+0-2}} \begin{bmatrix} l_b c\theta_0 - l_a c\theta_{+2-1-0} + l_0 c\theta_{+0-2} - h_0 s\theta_{+0-2} & 1 \end{bmatrix} \begin{bmatrix} F_z \\ M_\theta \end{bmatrix} \tag{5}$$

Substituting into formula (4) yields

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \dfrac{l_a l_0 s\theta_1 - l_a h_0 c\theta_1 + l_a l_b s\theta_{+1-2}}{l_b s\theta_0 + l_a s\theta_{+2-1-0} + l_0 s\theta_{+0-2} + h_0 c\theta_{+0-2}} & \dfrac{-l_a s\theta_{+2-1-0}}{l_b s\theta_0 + l_a s\theta_{+2-1-0} + l_0 s\theta_{+0-2} + h_0 c\theta_{+0-2}} \\ -\dfrac{l_b l_0 s\theta_2 - l_b h_0 c\theta_2 + l_b l_a s\theta_{+1-2}}{l_b s\theta_0 + l_a s\theta_{+2-1-0} + l_0 s\theta_{+0-2} + h_0 c\theta_{+0-2}} & \dfrac{-l_b s\theta_0}{l_b s\theta_0 + l_a s\theta_{+2-1-0} + l_0 s\theta_{+0-2} + h_0 c\theta_{+0-2}} \end{bmatrix} \begin{bmatrix} F_z \\ M_\theta \end{bmatrix} \tag{6}$$

where $\theta_0 = \theta + \theta_1$.

The subscripts f and r represent the front and hind legs, respectively, while A_f and A_r represent the foot-end coordinate systems of the front and hind legs, respectively. The quadruped model can be regarded as the parallel connection of four single-leg models. By combining these single-leg models, it

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

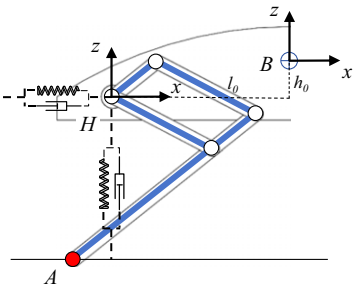


Figure 6. *Stance phase VMC.*

yields

$$\begin{bmatrix} \tau_{lf} \\ \tau_{rf} \\ \tau_{lr} \\ \tau_{rr} \end{bmatrix} = \begin{bmatrix} {}^{A_{lf}}_{B}J^T & & & \\ & {}^{A_{rf}}_{B}J^T & & \\ & & {}^{A_{lr}}_{B}J^T & \\ & & & {}^{A_{rr}}_{B}J^T \end{bmatrix} \begin{bmatrix} F_{lf} \\ F_{rf} \\ F_{lr} \\ F_{rr} \end{bmatrix} \tag{7}$$

3.2 Decomposed VMC model

In the global VMC, the overall virtual force F_{all} acting on the robot's body is considered as the final effective force. Consequently, within the developed quadruped model, F_{all} is utilized to substitute the vector $[\,F_{lf} \quad F_{rf} \quad F_{lr} \quad F_{rr}\,]^T$ in formula (7), yielding an 8×3 Jacobian-like matrix. This matrix facilitates the formulation of a mathematical relationship between the total virtual force and the driving joint torques associated with each leg. The resolution of this relationship necessitates operations involving an 8×8 matrix, thereby complicating the computational process. To address this complexity, the literature [34] introduces a decomposed VMC methodology, which segments the overarching control objective into distinct components for each leg based on specific principles. For each leg, corresponding models are developed for both the stance and swing phases. Through the coordinated control of each leg, the comprehensive control objective is ultimately realized.

The motion of the robot body primarily involves traction force in the X direction, support force in the Z direction, and the tilting angle θ around the center of mass. By analyzing the motion objectives along the X, Z, and θ directions, it becomes clear that the desired movements of the robot's body in these directions can be effectively attained through the precise regulation of force exerted by each leg in the X and Z directions.

Therefore, under the stance phase and the swing phase, corresponding virtual component models are constructed as shown in Figures 6 and 7. During the stance phase, the motion state of the body is controlled primarily through the interaction between the foot-end and the ground. The foot-end A is the external action point, and the hip H is the internal action point. Therefore, two sets of virtual spring-damping components in the X and Z directions are constructed at H. During the swing phase, the position of the foot-end is controlled mainly through the active joint torque of the hip. The hip H is the external action point, and the foot-end A is the internal action point. Therefore, two sets of virtual spring-damping components in the X and Z directions are constructed at A.

3.2.1 Stance phase

After decomposing the virtual forces required for the robot's motion, the virtual forces applied by each individual leg to the hip joint are obtained as

$$f = \begin{bmatrix} f_x \\ f_z \end{bmatrix} = \begin{bmatrix} k_x & \\ & k_z \end{bmatrix} \begin{bmatrix} x_d - x \\ z_d - z \end{bmatrix} + \begin{bmatrix} b_x & \\ & b_z \end{bmatrix} \begin{bmatrix} \dot{x}_d - \dot{x} \\ \dot{z}_d - \dot{z} \end{bmatrix} \tag{8}$$

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

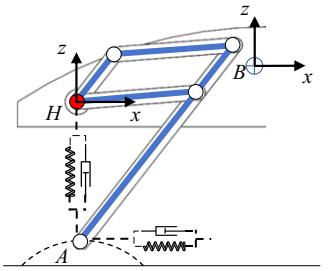


Figure 7. *Swing phase VMC.*

where k_x, k_z, b_x and b_z represent control gains (parameters of the virtual component); $x_d, z_d, \dot{x}_d$ and $\dot{z}_d$ represent the desired position and motion velocities of the robot platform.

To ensure motion stability, it is necessary to avoid fluctuations in the height of the robot body, thus setting $\dot{z}_d = 0$. z and $\dot{z}$ represent the real-time position and velocity of the foot-end relative to the hip coordinate system $\{H\}$ in the Z direction. x and $\dot{x}$ represent the displacement and measured velocity of the robot in the X direction. These values are derived from forward kinematics model.

The forward kinematics model of the foot-end relative to the hip coordinate system $\{H\}$ is

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} l_a \cos\theta_1 - l_b \cos\theta_2 \\ l_a \sin\theta_1 - l_b \sin\theta_2 \end{bmatrix} \tag{9}$$

By taking the partial derivatives of the joint angles, the Jacobian matrix is obtained as

$$^{H}_{A}J = \begin{bmatrix} -l_a s\theta_1 & l_b s\theta_2 \\ l_a c\theta_1 & -l_b c\theta_2 \end{bmatrix} \tag{10}$$

When a force is applied at the foot-end, the hip coordinate system serves as the reaction coordinate system. During the stance phase of the leg, a virtual force f needs to be applied at the hip. Consequently, applying f at the foot-end, the following can be derived:

$$\boldsymbol{\tau} = -^{H}_{P}J^{T} f \tag{11}$$

Expanding formula (11) yields

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = -\begin{bmatrix} -l_a s\theta_1 & l_a c\theta_1 \\ l_b s\theta_2 & -l_b c\theta_2 \end{bmatrix} \left(\begin{bmatrix} k_x & \\ & k_z \end{bmatrix} \begin{bmatrix} x_d - x \\ z_d - z \end{bmatrix} + \begin{bmatrix} b_x & \\ & b_z \end{bmatrix} \begin{bmatrix} \dot{x}_d - \dot{x} \\ \dot{z}_d - \dot{z} \end{bmatrix} \right) \tag{12}$$

3.2.2 Swing phase

The motion of the foot-end can be decomposed into two sets of spring-damper components along the X direction and Z direction of the hip coordinate system. The virtual force applied at the foot-end can be expressed as

$$f' = \begin{bmatrix} f'_x \\ f'_z \end{bmatrix} = \begin{bmatrix} k'_x & \\ & k'_z \end{bmatrix} \begin{bmatrix} x_d - x \\ z_d - z \end{bmatrix} + \begin{bmatrix} b'_x & \\ & b'_z \end{bmatrix} \begin{bmatrix} 0 - \dot{x} \\ 0 - \dot{z} \end{bmatrix} \tag{13}$$

In the application of VMC, there are no predetermined constraints regarding the trajectory of the foot's swing. The primary requirement is that the trajectory must fulfill the control objectives associated with the swing phase, specifically reaching the designated target position within the specified timeframe and elevating the leg sufficiently to avoid ground obstacles.

The virtual force is converted into joint torques through the single-leg Jacobian matrix, as

$$\boldsymbol{\tau}' = {}^{H}_{P}J^{T} f' \tag{14}$$

Expanding formula (14) yields

$$\begin{bmatrix} \tau'_1 \\ \tau'_2 \end{bmatrix} = \begin{bmatrix} -l_a s\theta_1 & l_a c\theta_1 \\ l_b s\theta_2 & -l_b c\theta_2 \end{bmatrix} \left(\begin{bmatrix} k'_x & \\ & k'_z \end{bmatrix} \begin{bmatrix} x_d - x \\ z_d - z \end{bmatrix} + \begin{bmatrix} b'_x & \\ & b'_z \end{bmatrix} \begin{bmatrix} 0 - \dot{x} \\ 0 - \dot{z} \end{bmatrix} \right) \tag{15}$$

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

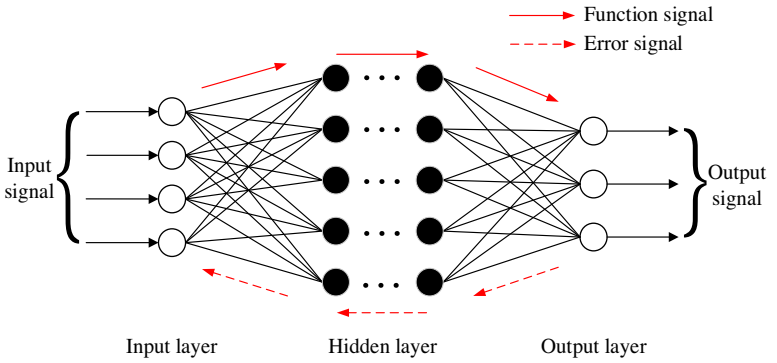


Figure 8. *Structure of BP neural network.*

Observing the established decomposed VMC models for the stance phase and swing phase, it becomes evident that the overall control effectiveness of these models is contingent upon the virtual parameter matrices K and B. In conventional VMC algorithms, the values assigned to the parameters within the K and B matrices are typically derived through empirical trial-and-error approaches. Once these values are established, they remain static and are not subject to modification. However, during the execution of motion, various factors, including alterations in the motion state and variations in terrain conditions, can significantly influence the performance of the controller. The rigidity of these parameters consequently limits the adaptability and robustness of the controller, making it challenging to ensure optimal control outcomes. To mitigate this limitation, this paper proposes the integration of a BPNN to facilitate the adaptive adjustment of the parameters within the K and B matrices. This approach aims to enhance the adaptive capabilities of the controller, thereby improving both the control accuracy and dynamic performance of the VMC controller across diverse motion conditions.

4. BP-VMC system

4.1 BP neural network

The BPNN [35], as illustrated in Figure 8, comprises an input layer, one or more hidden layers, and an output layer. Each neuron within the network is interconnected with all neurons in the preceding layer. The process of signal transmission occurs in two primary phases: forward propagation and backward propagation. During forward propagation, function signals traverse through the layers until they arrive at the output layer, where the output value is determined based on the synaptic weights that exist between the neurons. An error signal is produced by evaluating the discrepancy between the network's output value and the intended output value. This error signal is subsequently propagated backward through the layers until it reaches the input layer. By modifying the weight matrices between each layer, the network aims to minimize the error, progressively aligning with the training objective.

4.1.1 Transmission and calculation of signals

The input and output of the BPNN's input layer in this paper are

$$\begin{cases} I_i^{\mathbf{0}}(n) = x_i(n) \\ O_i^{\mathbf{0}}(n) = I_i^{\mathbf{0}}(n) \end{cases} \quad i = 1, 2, \cdots, M \tag{16}$$

where the superscript represents the network layer corresponding to the signal, with 0, 1, and 2 referring to the input layer, hidden layer, and output layer, respectively. The subscript i denotes the node number in the input layer, and M represents the total number of nodes in the input layer.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

The input variables for the VMC controller are

$$\begin{aligned} x_1 &= e(n) \\ x_2 &= e(n) - e(n-1) \end{aligned} \tag{17}$$

where ***e(n)*** represents the control system error

$$e(n) = x_{in}(n) - y_{out}(n) \tag{18}$$

The induced local field and output of the hidden layer in the neural network are, respectively

$$\begin{cases} I_j^1 = \sum_{i=0}^{M} O_i^0(n) w_{ij}^1(n) \\ O_j^1 = h(I_j^1) \end{cases} \quad j = 1, 2, \cdots, Q \tag{19}$$

where w_{ij}^1 represents the synaptic weights of the hidden neurons and Q is the number of nodes in the hidden layer. The activation function for the hidden neurons is chosen as the hyperbolic tangent function.

$$h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{20}$$

The induced local field and output of the output layer in the neural network are, respectively

$$\begin{cases} I_l^2 = \sum_{j=0}^{Q} O_j^1(n) w_{jl}^2(n) \\ O_l^2 = g(I_l^2) \end{cases} \quad l = 1, 2, \cdots, T \tag{21}$$

where w_{jl}^2 represents the synaptic weights of the output neurons and T is the number of nodes in the output layer. The output of each node in the output layer corresponds to the target training parameter. Since the parameter values $\boldsymbol{K}$ and $\boldsymbol{B}$ in this paper are non-negative, the activation function of the output layer is chosen as a non-negative *sigmoid* function.

$$g(x) = \frac{e^x}{e^x + e^{-x}} \tag{22}$$

Simultaneously, due to the output range of the activation function being [0, 1], the weight matrix $\boldsymbol{P}$ for the VMC parameters is defined and incorporated into the control law of the VMC controller, resulting in

$$u(n) = \sum_{l=1}^{T} P_l O_l^2(n) x_l(n) \tag{23}$$

Substitute into the quadruped robot model to obtain the final output value.

$$y_{out}(n) = F_{VMC}(u(n)) \tag{24}$$

4.1.2 Neural network training algorithm

Using the BP learning algorithm, the synaptic weights of the neural network are iteratively corrected. The cost function of the system is defined as

$$\boldsymbol{\varepsilon}(n) = \frac{1}{2} e^2(n) = \frac{1}{2} \left(x_{in}(n) - y_{out}(n) \right)^2 \tag{25}$$

To adjust the synaptic weights of the network according to the gradient descent method, the adjustment is performed by searching in the direction of the negative gradient of the synaptic weight coefficients with respect to $\boldsymbol{\varepsilon}(n)$, and an inertia term is added to enable faster convergence of the search. It yields

$$\boldsymbol{\Delta} w_{jl}^2(n) = -\eta \frac{\partial \boldsymbol{\varepsilon}(n)}{\partial w_{jl}^2(n)} + \alpha \boldsymbol{\Delta} w_{jl}^2(n-1) = \eta \delta_l^2(n) O_j^1(n) + \alpha \boldsymbol{\Delta} w_{jl}^2(n-1) \tag{26}$$

where η is learning rate and α is momentum factor.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

According to the chain rule of differentiation, the local gradient $\boldsymbol{\delta}_l^2(n)$ can be expressed as

$$\boldsymbol{\delta}_l^2(n) = -\frac{\partial \boldsymbol{\varepsilon}(n)}{\partial I_l^2(n)} = -\frac{\partial \boldsymbol{\varepsilon}(n)}{\partial y_{out}(n)} \frac{\partial y_{out}(n)}{\partial u(n)} \frac{\partial u(n)}{\partial O_l^2(n)} \frac{\partial O_l^2(n)}{\partial I_l^2(n)} \tag{27}$$

As shown in formula (24), $\frac{\partial y_{out}(n)}{\partial u(n)}$ is associated with the specific model of the control system. If accurately calculated during each iteration of optimization, it will significantly increase the computational complexity and potentially affect the response speed of the controller. Therefore, $\frac{\partial y_{out}(n)}{\partial u(n)}$ is approximated as a sign function $\operatorname{sgn}\left(\frac{\partial y_{out}(n)}{\partial u(n)}\right)$, and the errors resulting from this simplified calculation can be compensated for by adjusting the learning rate appropriately.

Substituting formulas (21) and (23) into (27), it yields

$$\boldsymbol{\delta}_l^2(n) = e(n)\operatorname{sgn}\left(\frac{\partial y_{out}(n)}{\partial u(n)}\right) x_l(n) g'\left(I_l^2(n)\right) \tag{28}$$

Therefore, the synaptic weight adjustment formula for the neurons in the output layer of the network is

$$\boldsymbol{\Delta} w_{jl}^2(n) = \eta \boldsymbol{\delta}_l^2(n) O_j^1(n) + \alpha \boldsymbol{\Delta} w_{jl}^2(n-\mathbf{1}) \tag{29}$$

Similarly, the synaptic weight adjustment formula for the neurons in the hidden layer is

$$\boldsymbol{\Delta} w_{ij}^1(n) = \eta \boldsymbol{\delta}_j^1(n) O_i^0(n) + \alpha \boldsymbol{\Delta} w_{ij}^1(n-\mathbf{1}) \tag{30}$$

where $\boldsymbol{\delta}_j^1(n)$ represents the local gradient for the neurons in the hidden layer, as

$$\boldsymbol{\delta}_j^1(n) = h'\left(I_j^1(n)\right) \sum_{l=1}^{T} \boldsymbol{\delta}_l^2(n) \boldsymbol{\Delta} w_{jl}^2(n) \tag{31}$$

4.1.3 BPNN structure parameters

As indicated in Sections 4.1.1 and 4.1.2, the performance of the BPNN is notably influenced by the number of hidden layers and neurons utilized. The VMC control system of the quadruped robot, where the BPNN is applied in this paper, is a complex continuous system with multiple inputs and multiple outputs. To achieve optimal real-time performance of the control system during operation, it is imperative that the BPNN not only provides effective function approximation but also minimizes structural complexity and improves learning efficiency.

Robert Hecht-Nielsen has proven that any continuous function within a closed interval can be approximated by a BP network with one hidden layer [36]. Consequently, the number of hidden layers is selected as 1.

The quantity of hidden neurons influences the performance of the neural network in terms of fitting. Having too few neurons may cause "underfitting," whereas having too many can lead to "overfitting" and a higher computational burden [37]. In this paper, the BPNN is designed with four input nodes (comprising three input variables and one bias) and two output nodes (representing the output values). To reduce the likelihood of "overfitting," the guideline for choosing the number of hidden neurons is to keep it below twice the size of the input layer. After thorough analysis and simulation, the final decision is made to set the number of hidden neurons at 5.

Therefore, the final BPNN structure is established as "$4 \times 5 \times 2$."

4.2. BP-VMC controller structure

Neural networks represent a category of machine learning techniques that emulate the operational principles of biological neural systems, demonstrating capabilities for self-learning, self-organization, and adaptability. In the context of control systems, these networks facilitate adaptive control for complex systems characterized by uncertainties, significant nonlinearities, temporal variations, and delays.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

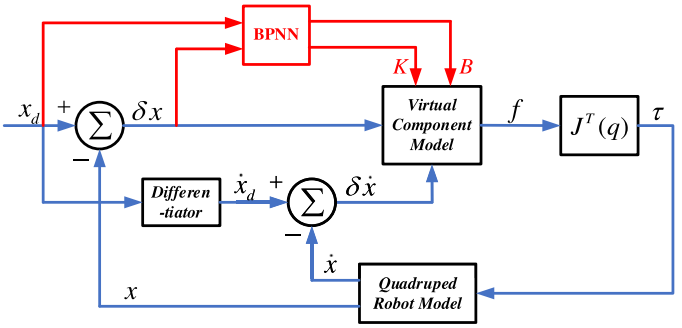


Figure 9. *Schematic diagram of BP-VMC control principle.*

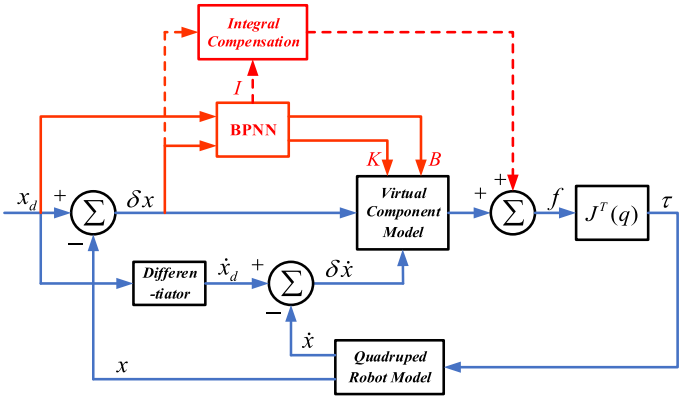


Figure 10. *Schematic diagram of BP-VMC with IC.*

Since the essence of the VMC algorithm is to adjust the control input based on the error between the desired state and the actual state. Similarly, the BPNN employs error signals to inform the adjustment of weights and biases within neuron nodes, utilizing the backward propagation of these error signals to minimize output error. Therefore, combining VMC with BPNN and training the virtual parameter matrices K and B through the backpropagation algorithm enables adaptive adjustment of the VMC controller, allowing it to accommodate various systems and environments. The schematic diagram of the BP-VMC control system proposed in this paper is shown in Figure 9.

The control law of the BP-VMC controller can be expressed as

$$\begin{aligned} \boldsymbol{\tau} &= J^T f = J^T \left(K(n)\boldsymbol{\delta} x + B(n)\boldsymbol{\delta}\dot{x}\right) \\ \begin{bmatrix} K(n) \\ B(n) \end{bmatrix} &= BPNN\ (x_d, \boldsymbol{\delta} x) \end{aligned} \tag{32}$$

During the development of the motion controller, the inherent mass characteristics of the robot are often neglected, leading to notable deviations in z-axis height control. A thorough analysis of the control law reveals that the VMC resembles to a PD controller, posing challenges in mitigating these deviations. To address this issue, the integration of an error compensation term into the z-axis controller is proposed to counterbalance the robot's self-weight. However, given the variations in compensation requirements caused by differing motion states and external disturbances, the integral compensation (IC) method is adopted. This methodology accumulates errors and adjusts the control output accordingly, thereby eliminating steady-state errors and enhancing the precision of motion control. Additionally, the integral coefficient within the integral compensator can be autonomously tuned using the BPNN, negating the need for manual parameter adjustments. Therefore, based on the block diagram in Figure 9, the BP-VMC control block diagram with IC is obtained as shown in Figure 10.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

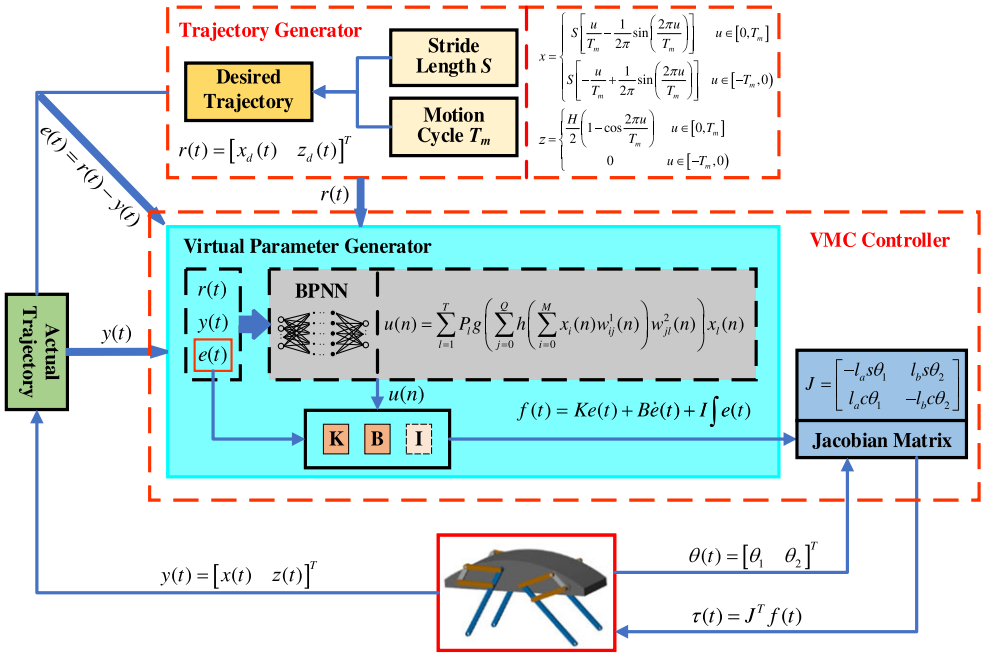


Figure 11. *BP-VMC controller scheme.*

The control law of the BP-VMC with IC controller is

$$\begin{aligned} \boldsymbol{\tau} &= J^T f = J^T \left(K(n)\delta x + B(n)\delta \dot{x} + I(n) \int \delta x dt\right) \\ \begin{bmatrix} K(n) \\ B(n) \\ I(n) \end{bmatrix} &= BPNN\ (x_d, \delta x) \end{aligned} \tag{33}$$

5. BP-VMC controller scheme

The quadruped robot controller scheme is shown in Figure 11, which mainly includes a trajectory generator, VMC controller, and quadruped robot model. Initially, the trajectory generator generates the desired trajectory curve based on the preset velocity and periodicity conditions, outputting the corresponding desired position of the foot-end in real time. Subsequently, the actual current position of the foot-end obtained from the quadruped robot model is compared with the desired position of the foot-end, and the resulting error signal is transmitted to the virtual parameter generator. In the virtual parameter generator, the BPNN modifies and optimizes the virtual parameters based on the error signal and calculates the corresponding virtual force to be passed to the Jacobian matrix. Ultimately, the Jacobian matrix combines the driving joint angle signal obtained from the quadruped robot model and the received virtual force signal to calculate the corresponding joint driving torque, which drives the quadruped robot model to achieve the corresponding movement, resulting in a new foot-end position signal. This new signal is then compared with the desired foot-end position to obtain a new error signal, initiating the next control cycle.

The trajectory generator generates corresponding foot trajectory curves based on the desired velocity and motion cycle, guiding the quadruped robot to move according to the desired motion state. The velocity of the quadruped robot is closely related to its gait cycle and stride length. Therefore, controlling the stride length and stride frequency of the quadruped robot in the direction of travel can achieve the purpose of velocity control. As a result, the trajectory planning formulas for the trot gait at the foot-ends

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

Table I. *Simulation parameter settings.*

Parameter	Value
l_1, l_2, l_3, l_4	0.24 m, 0.12 m, 0.24 m, 0.4 m
L_B, W_B, H_B	1 m, 0.3 m, 0.2 m
M_B, M_1, M_2, M_3, M_4	20 kg, 1 kg, 1 kg, 1 kg, 2 kg
(VMC) k_x, k_z	5600 N/m, 1.51×10^4N/m
(VMC) b_x, b_z	1300 N/(m/s), 2560 N/(m/s)
Initial height H_0, lifting height H	0.35 m, 0.08 m
(BPNN) learning rate η, momentum factor α	0.8, 0.3

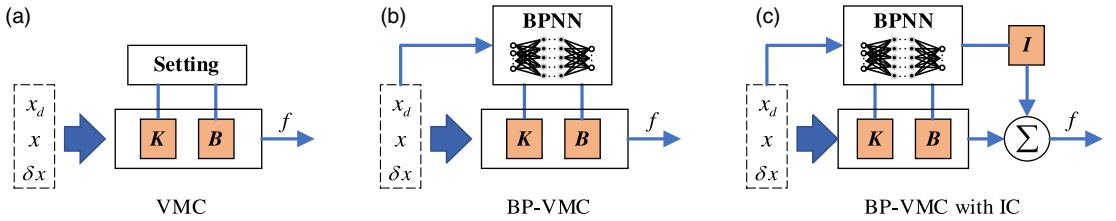


Figure 12. *Virtual parameter generator.*

are shown in formulas (34) and (35).

$$x = \begin{cases} S\left[\frac{u}{T_m} - \frac{1}{2\pi}\sin\left(\frac{2\pi u}{T_m}\right)\right] & u \in [0, T_m] \\ S\left[-\frac{u}{T_m} + \frac{1}{2\pi}\sin\left(\frac{2\pi u}{T_m}\right)\right] & u \in [-T_m, 0) \end{cases} \tag{34}$$

$$z = \begin{cases} \frac{H}{2}\left(1 - \cos\frac{2\pi u}{T_m}\right) & u \in [0, T_m] \\ 0 & u \in [-T_m, 0) \end{cases} \tag{35}$$

where T_m represents the swing phase period, and in the trot gait, $T_m = 1/2T$. u is the given periodic time signal, $u \in [-T_m, T_m]$. The stride length S is determined by the tracking velocity, $S = v_{xd}T_m$. H represents the desired leg lift height, usually $H = constant$.

The VMC controller is mainly composed of a Jacobian matrix and a virtual parameter generator. The Jacobian matrix reflects the mapping relationship between the joint actuators and the end-effectors, which is determined by the specific structure of the quadruped robot. The main role of the virtual parameter generator is to determine the parameter values of the virtual components and calculate the output virtual force at the foot-end. Three types of virtual parameter generators are designed (Figure 12), which, respectively, achieve the output of virtual parameters in the form of fixed values, BPNN training, and BPNN training with IC.

6. Simulations

6.1 Building of simulation system

In order to assess the control accuracy and stability of the BP-VMC algorithm, a simulation control system was developed based on the control framework diagram (refer to Figure 11) for conducting simulation experiments. The models of the quadruped robot and the ground were created utilizing the Simscape Multibody module, wherein the kinematic joints and physical interactions were specified. The controller module was implemented in Simulink. The simulation environment and control system for the quadruped robot, as illustrated in Figures 13 and 14, comprise a trajectory generator, a VMC controller, a single-leg model, and a body model. The VMC controller is further categorized into a Jacobian matrix and a virtual parameter generator. The virtual parameter generator was configured in three distinct forms: fixed value (Figure 15(a)), BP network training generation (Figure 15(b)), and BP training with IC (Figure 15(c)) for comparative simulation experiments. The simulation parameters are

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

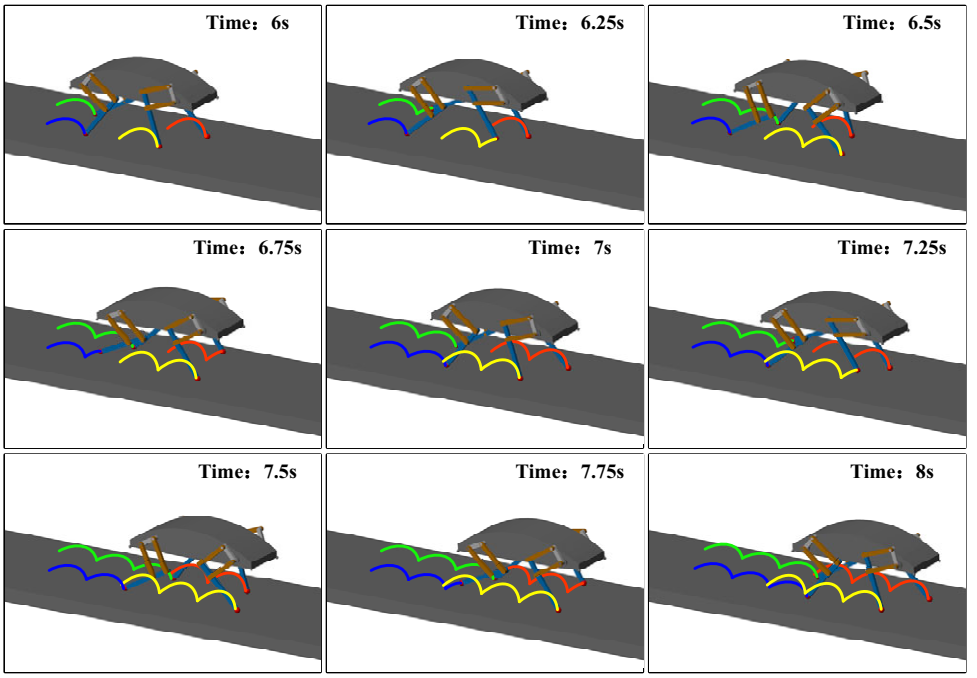


Figure 13. *Quadruped robot simulation environment.*

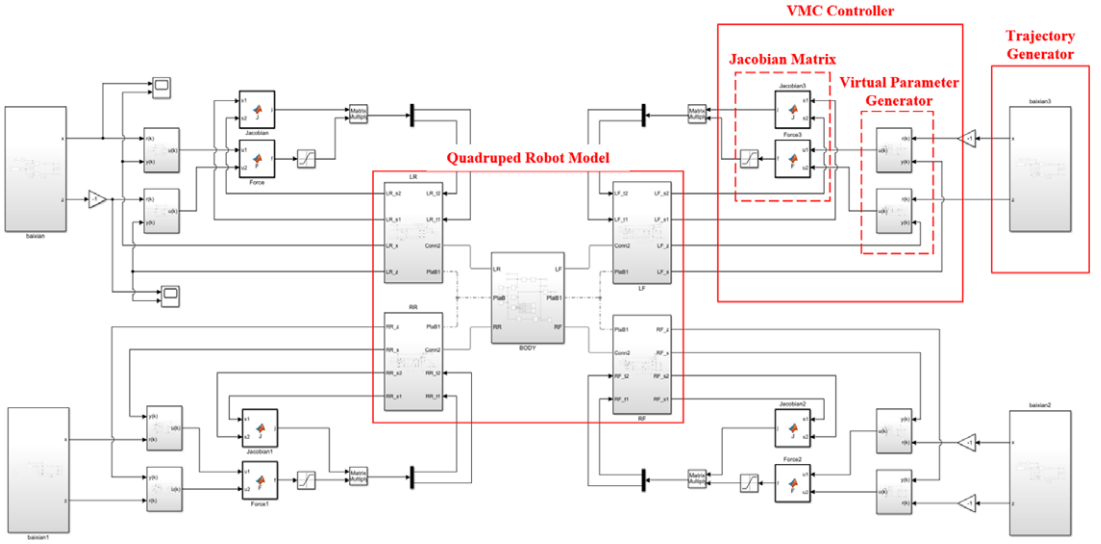


Figure 14. *Simulation model of control system.*

detailed in Table I. The fixed parameter values of the VMC are established as the self-tuning parameter values of the BP-VMC during the initial 4 s of the simulation, thereby ensuring a consistent initial performance of the controller.

6.2 Trajectory tracking

The simulation analysis was conducted using the trot gait as the primary motion gait for the quadruped robot. The complete gait cycle time was set to $T = 1$ s, with the swing phase and support phase each

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

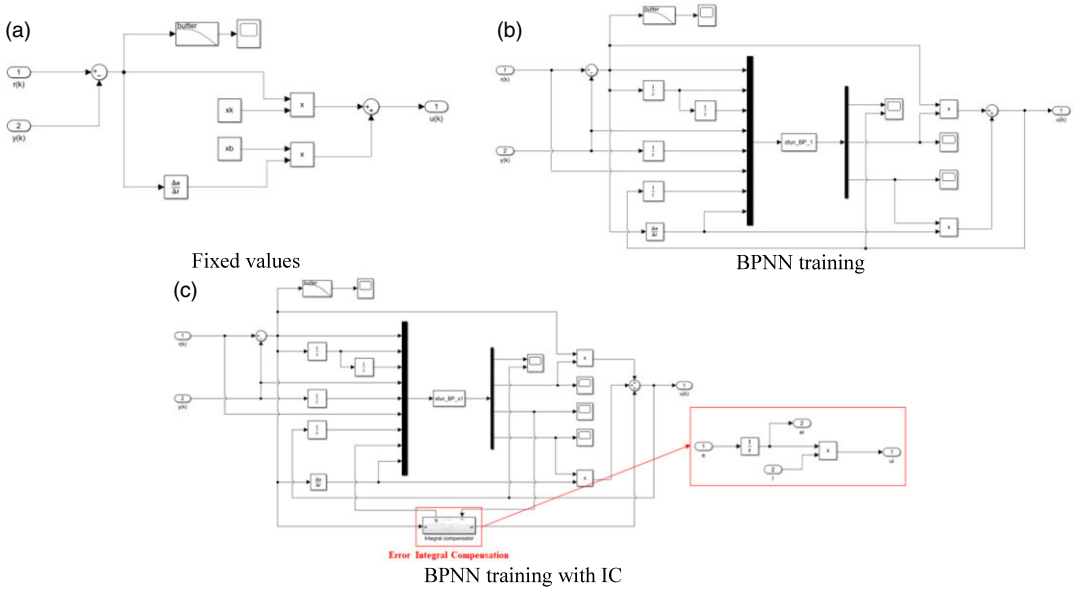


Figure 15. *Simulation model of virtual.*

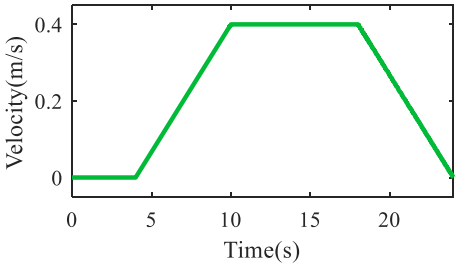


Figure 16. *Desired velocity.*

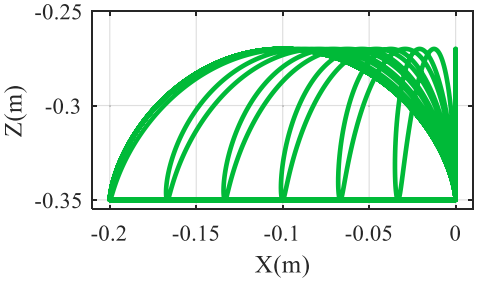


Figure 17. *Desired foot trajectory.*

accounting for 50% of the motion cycle [38], that is, $T_{sw}=T_{st}=0.5$ s. The desired tracked motion velocity curve and foot-end trajectory are shown in Figures 16 and 17. The main motion process is as follows: the robot remains stationary in the quadrupedal stance for the first 4 s to enter an initial equilibrium state, ensuring consistency in the initial state; from 4 to 10 s, the robot gradually accelerates from 0 to 0.4 m/s; from 10 to 18 s, the robot maintains a constant velocity of 0.4 m/s; and from 18 to 24 s, the robot gradually decelerates from 0.4 m/s to 0.

Simulations and comparative experiments were conducted using both VMC and BP-VMC methods. The parameter settings for traditional VMC are shown in Table I, and the parameter values for BP-VMC are presented in Figures 20 and 23. According to formulas (8) and (13), the virtual forces and

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

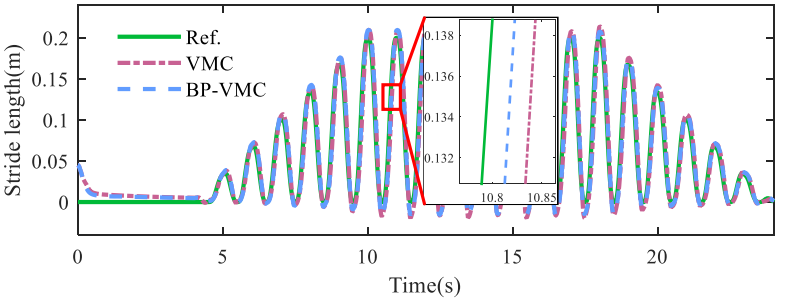


Figure 18. *Stride length in the X direction.*

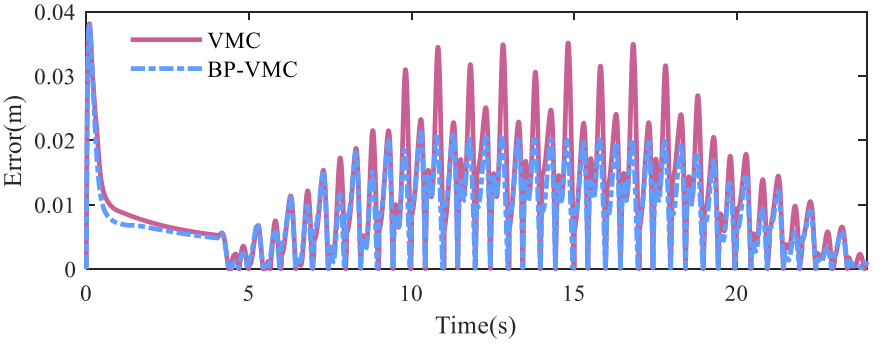


Figure 19. *Absolute error in the X direction.*

virtual parameters in the X and Z directions are independent. Therefore, the simulation results in the X and Z directions were analyzed separately, and the final comparison was made based on the foot-end trajectories throughout the entire simulation process. Since the first 4 s are for adjusting the robot's initial equilibrium state, the subsequent error analysis uses data obtained after this 4-s simulation.

6.2.1 X direction

An examination of Figure 18 reveals that the BP-VMC exhibits superior real-time tracking performance compared to the VMC, aligning more closely with the desired trajectory. To quantitatively assess the tracking efficacy, the error curve presented in Figure 19 has been generated. At lower velocities, the tracking errors for both controllers are comparable, with BP-VMC demonstrating a marginally reduced error relative to VMC. However, as the velocity escalates, the error values for both controllers increase, with a notable abrupt change in the error curve for VMC, which reaches a maximum error of 3.470×10^{-2} m. In contrast, BP-VMC maintains a maximum error of only 1.997×10^{-2} m. Throughout the entire evaluation period, the root mean square error (RMSE) for VMC tracking is recorded at 1.356×10^{-2}, whereas the RMSE for BP-VMC tracking is 1.019×10^{-2}, representing 75.15% of the RMSE associated with VMC.

Based on the simulation results in Figures 18 and 19, along with the K_X parameter variation curve of BP-VMC in Figure 20, it can be observed that during the swing phase, the K_X value increases, leading to an increase in the controller's stiffness. This results in faster tracking response and smaller tracking errors for BP-VMC. Furthermore, by analyzing the B_X parameter variation curve of BP-VMC in Figure 19, it is found that at the moment when there is a sudden error change in VMC, the B_X parameter of BP-VMC increases, which increases the damping coefficient of the controller. This effectively suppresses the occurrence of sudden error changes and enhances the control stability.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

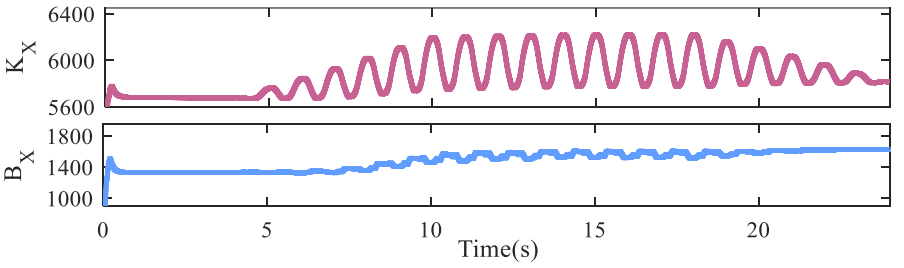


Figure 20. *BP-VMC parameters in the X direction.*

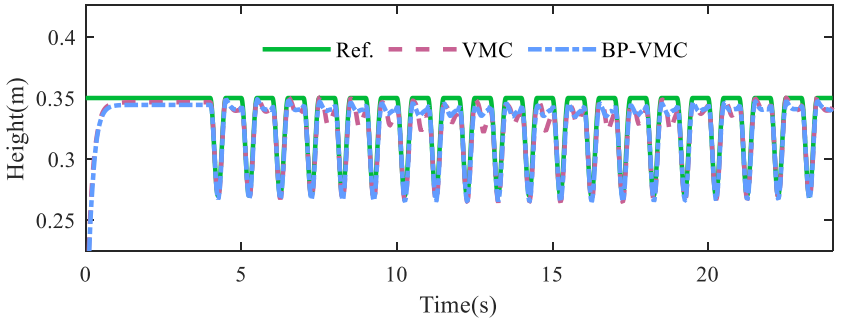


Figure 21. *Height between the foot-end and the center of mass.*

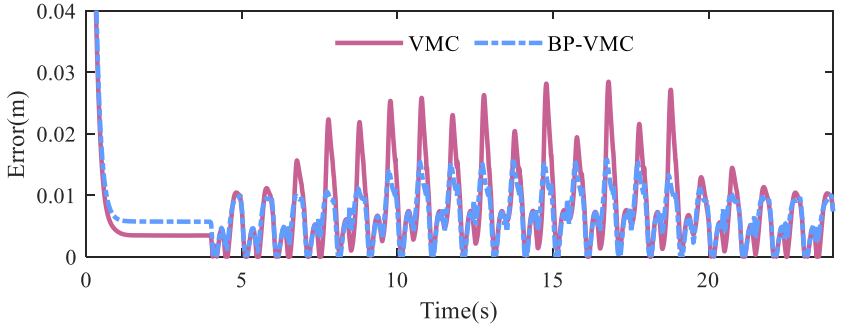


Figure 22. *Absolute error in the Z direction.*

6.2.2 Z direction

The results of height control in the Z direction are presented in Figure 21. Overall, both the VMC and the BP-VMC demonstrate the capability to accurately track the reference values. During the swing phase, the control performance of both VMC and BP-VMC is comparable, as they closely align with the reference curve, indicating effective tracking capabilities. However, during the support phase, notable discrepancies arise in the control performance of the two methods. The transition between phases – specifically from bipedal to quadrupedal and back to bipedal support – induces fluctuations in the height of the robot's center of mass. This phenomenon is clearly illustrated in the curve depicted in Figure 21.

To further quantify the tracking performance, the error curve illustrated in Figure 22 is obtained. The maximum error value of VMC is 2.852×10^{-2} m, with RMSE of 9.848×10^{-3}. The maximum error value of BP-VMC is 1.595×10^{-2} m, with RMSE of 7.584×10^{-3}. The RMSE of BP-VMC is 77.01% of VMC's RMSE, and its maximum error value is significantly smaller than VMC's, indicating that BP-VMC has higher tracking accuracy.

Analysis of the fluctuation trends in the respective curves indicates that, at lower velocities, the height fluctuations of VMC and BP-VMC exhibit a high degree of consistency. However, as the velocity escalates, the height fluctuations associated with VMC experience a marked increase, characterized by

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

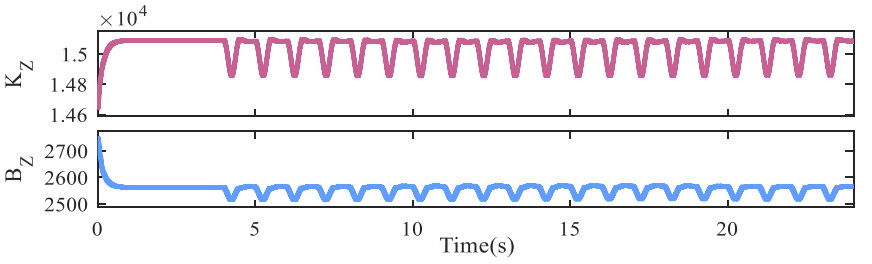


Figure 23. *BP-VMC parameters in the Z direction.*

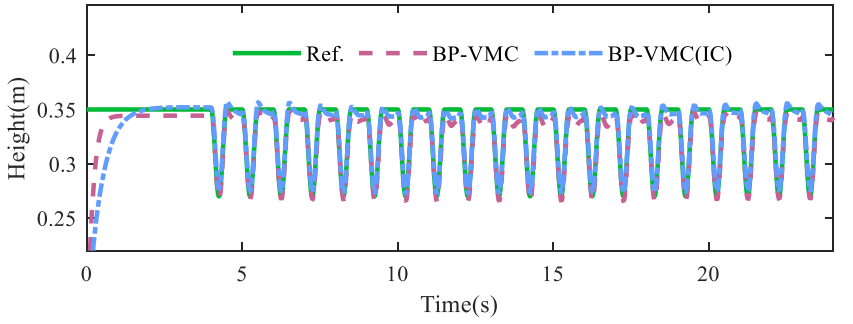


Figure 24. *Height of BP-VMC with IC.*

pronounced peaks. Conversely, the height fluctuations of BP-VMC remain relatively stable, showing minimal variation from low-velocity conditions and lacking abrupt changes in response to alterations in the motion state. Consequently, the comparative results suggest that BP-VMC exhibits superior control stability and adaptability in contrast to VMC, thereby facilitating consistent motion state control across varying motion conditions.

Furthermore, it is evident that a significant steady-state error exists in the height control within the Z direction. This is due to the neglect of the robot's self-weight in the process of virtual model construction, and there is no integral term to compensate for the steady-state error. Consequently, an integral compensator has been incorporated into the Z direction controller to rectify the steady-state error attributable to the robot's self-weight. The simulation outcomes following the implementation of this compensation are illustrated in Figure 24. While the introduction of the error integral compensator does influence the response speed of the controller, it effectively mitigates the steady-state error, thereby significantly enhancing the precision of height tracking control in the Z direction.

6.2.3 Foot trajectory

During the simulation process, the stride length of the robot's foot movement varies with the change in tracking velocity, achieving velocity tracking. Based on the above simulation results, the foot trajectory curves of VMC and BP-VMC are plotted separately, as shown in Figure 25.

Observing the foot trajectory curve of VMC, while its overall shape resembles the desired trajectory curve, there are significant discrepancies in the precise numerical tracking. The curve overall has a noticeable offset compared to the reference curve, indicating a relatively obvious control error in the motion control. Additionally, during the support phase, there are pronounced fluctuations in the height of the foot-end, which leads to significant differences in the lower part of the VMC foot trajectory curve compared to the reference curve.

The foot trajectory curve of BP-VMC demonstrates a substantial degree of alignment with the desired trajectory curve. Throughout both the swing and support phases, the curve exhibits minimal deviation from the desired trajectory curve. In comparison to the foot trajectory curve of VMC, BP-VMC maintains

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

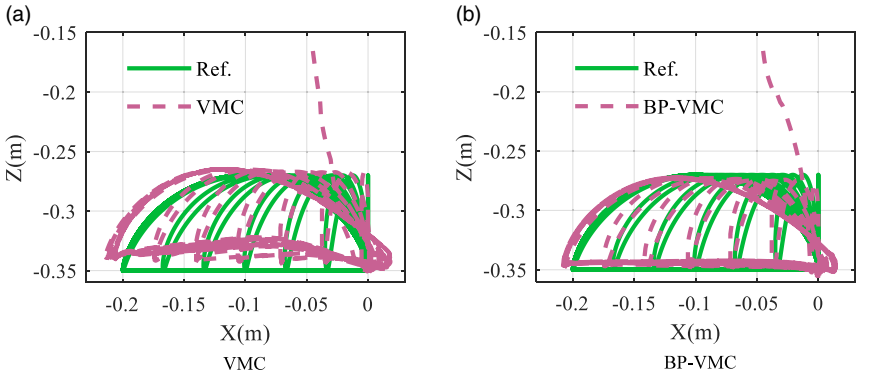


Figure 25. *Foot-end movement trajectory.*

a more stable foot height during the support phase, without obvious height fluctuations, and shows a greater consistency with the desired trajectory curve.

To more intuitively compare the trajectory tracking accuracy of the two controllers, the average geometric distance between the corresponding points on the simulation result curve and the reference trajectory curve is taken as the average tracking error.

$$\overline{D} = \frac{\sum_{i=0}^{n} \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2}}{n} \tag{36}$$

where (x_i, y_i) represents the point on the simulation result curve and (X_i, Y_i) represents the point on the reference trajectory curve.

Based on this, the average tracking errors for VMC and BP-VMC are, respectively, obtained as follows:

$\overline{D}_{VMC} = 1.448 \times 10^{-2}, \overline{D}_{BP-VMC} = 0.974 \times 10^{-2}\ (m)$

The tracking error of BP-VMC is 67.27% of VMC's. Combining the comparison of the curve results in Figure 25, it can be seen that the control accuracy and stability of BP-VMC are significantly better than VMC, achieving more precise foot trajectory tracking control.

6.3 Velocity tracking

In addition to trajectory tracking, velocity tracking constitutes a vital aspect of motion control in quadruped robots. In the velocity tracking simulation experiment, the velocity curve depicted in Figure 26 represents the desired tracking velocity, which incorporates various velocity values and working conditions during both the acceleration and deceleration phases. The resultant centroid velocity curve is depicted in Figure 27. During the trotting gait, the two diagonal pairs of legs alternate between supporting and swinging phases. This transition between legs results in fluctuations in the centroid velocity, due to the reaction force generated when the swinging leg makes contact with the ground.

To facilitate a more intuitive comparison between the velocity curve from the robot and the desired motion velocity curve, the centroid velocity curve is subjected to smoothing. By designating one swing phase cycle as the sampling period, the simulation time is partitioned into n cycles, during which the displacement of the robot is documented for each cycle. This methodology enables the calculation of the average velocity for each cycle, yielding a sequence of discrete velocity points arranged in chronological order. Subsequently, these points are interconnected with a smooth curve to effectively illustrate the velocity curve throughout the robot simulation process.

$$t = T_{SW} \quad , \quad n = \frac{T}{t}$$

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

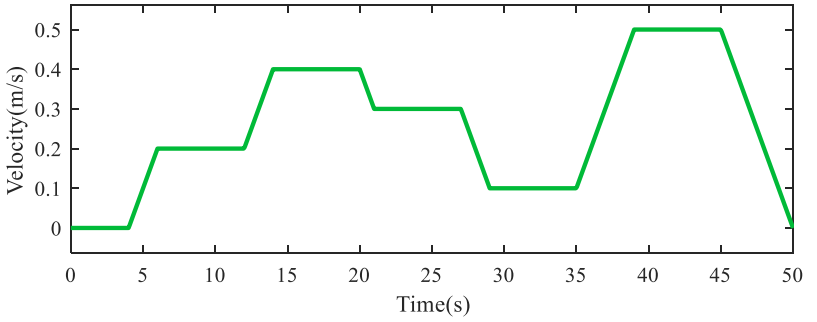


Figure 26. *The desired velocity curve for velocity tracking.*

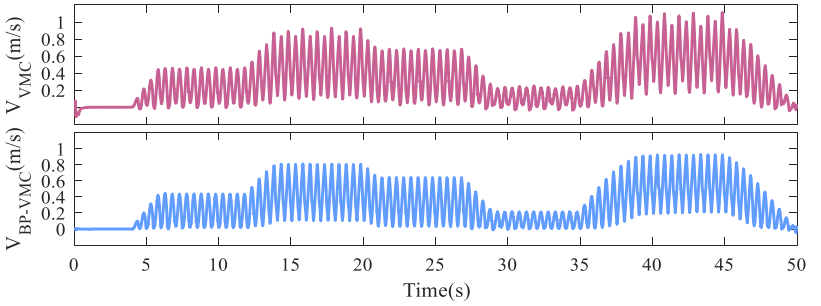


Figure 27. *Centroid forward velocity in the direction of motion.*

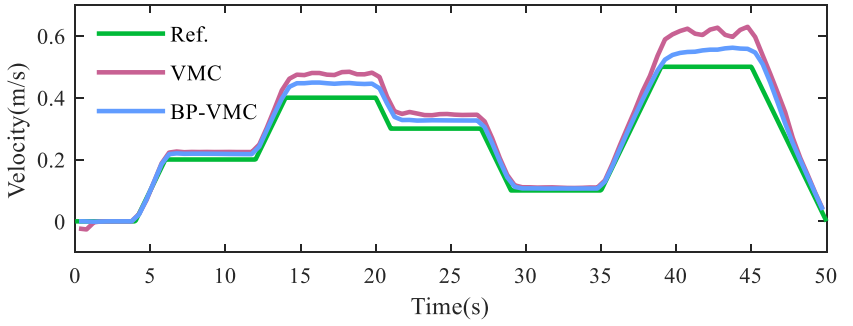


Figure 28. *Average centroidal velocity within cycle.*

$$\begin{cases} \Delta x_i = X(i) - X(i-1) \\ \bar{v}_i = \frac{\Delta x_i}{t} \end{cases} \qquad i = 1, 2, \ldots, n \tag{37}$$

where T is the total simulation time and $X(i)$ is the position of the robot recorded during the ith cycle.

Consequently, the centroid velocity curves and error curves presented in Figures 28 and 29 have been derived. At tracking velocities of 0.1 and 0.2 m/s, both VMC and BP-VMC demonstrate the capability to sustain velocity tracking with tiny errors and stable outputs. However, as the tracking velocity exceeds 0.3 m/s, there is a noticeable increase in tracking error values, accompanied by significant fluctuations in the velocity curve of VMC. In contrast, BP-VMC continues to exhibit a relatively stable velocity output, thereby indicating its superior control stability in comparison to VMC. This suggests that BP-VMC is more effective in maintaining a consistent velocity output across varying tracking conditions.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

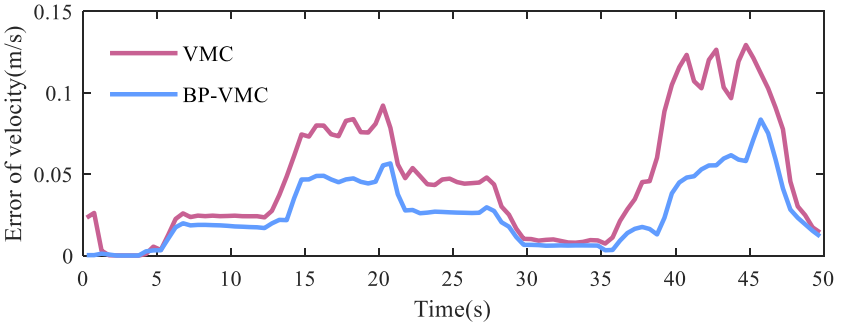


Figure 29. *Velocity tracking absolute error.*

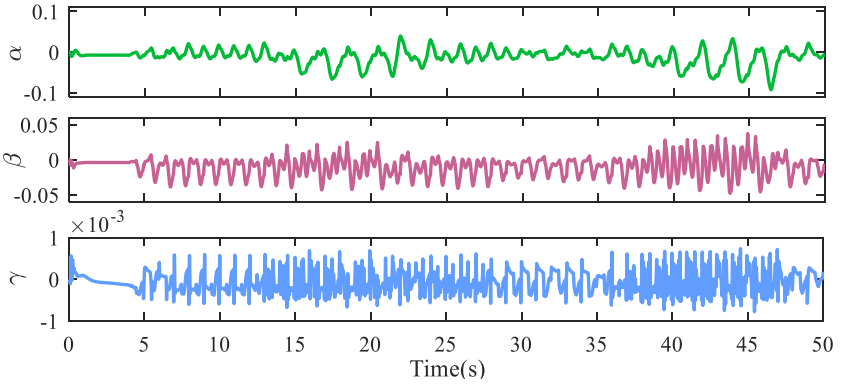


Figure 30. *VMC attitude angle.*

To more intuitively compare the accuracy of velocity tracking errors between VMC and BP-VMC, the RMSE is used to quantify their velocity tracking errors.

$$R = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \quad i = 1, 2, \ldots, n \tag{38}$$

The calculated RMSE for VMC and BP-VMC are, respectively:

$$R_{VMC} = 5.914 \times 10^{-2}, R_{BP-VMC} = 3.278 \times 10^{-2}$$

The error value of BP-VMC is 55.43% of VMC's, indicating that BP-VMC's velocity tracking accuracy is optimized by 44.57% compared to VMC.

6.4 Stability analysis

The stability margin serves as a metric for evaluating a robot's stability in a static condition; however, it is inadequate for assessing stability during locomotion. The stability of the robot's body while in motion is linked to its attitude angles (α, β, γ) [39]. Consequently, the instability vector $\mathbf{\Delta}$ can be defined as the absolute deviation between the actual attitude angles of the robot's body and the target angles. The instability associated with the robot's movement can subsequently be quantified by computing the L2 norm of $\mathbf{\Delta}$.

$$\mathbf{\Delta} = \left(|\alpha_i - \alpha_d|, \quad |\beta_i - \beta_d|, \quad |\gamma_i - \gamma_d| \right) \tag{39}$$

$$\|\mathbf{\Delta}\|_2 = \sqrt{|\alpha_i - \alpha_d|^2 + |\beta_i - \beta_d|^2 + |\gamma_i - \gamma_d|^2} \tag{40}$$

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

Table II. *Range of attitude angle fluctuations.*

	Roll angle α	**Pitch angle β**	**Yaw angle γ**
VMC	$-0.09 \sim 0.04$ rad	$-0.05 \sim 0.04$ rad	$-8 \times 10^{-4} \sim 8 \times 10^{-4}$ rad
BP-VMC	$-0.05 \sim 0.05$ rad	$-0.03 \sim 0.03$ rad	$-8 \times 10^{-4} \sim 8 \times 10^{-4}$ rad

Table III. *Mean value of attitude angle absolute deviation and instability metric.*

Mean value	**VMC**	**BP-VMC**
Roll angle absolute error	1.596×10^{-2}	1.184×10^{-2}
Pitch angle absolute error	1.255×10^{-2}	0.764×10^{-2}
Yaw angle absolute error	2.059×10^{-4}	2.011×10^{-4}
Instability metric $\|\Delta\|_2$	2.278×10^{-2}	1.500×10^{-2}

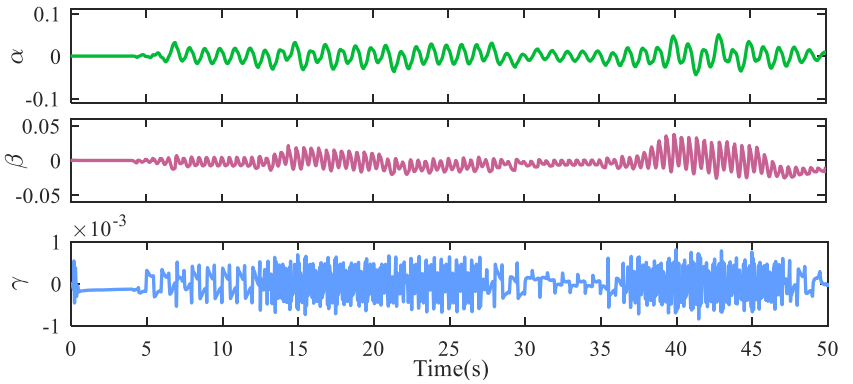


Figure 31. *BP-VMC attitude angle.*

Utilizing the velocity curve presented in Figure 26 for simulation purposes, the attitude angle curves for the VMC and BP-VMC control simulations are obtained as shown in Figures 30 and 31. The fluctuation ranges of the attitude angles are also listed in a table format (Table II).

Based on the observed trend in the curve, it is evident that as velocity increases, the fluctuation range of the VMC also expands, exhibiting pronounced asymmetric fluctuations. This suggests that there is a certain difference in the control of the four legs, which leads to instability in the motion state. In contrast, the attitude angle fluctuations of the BP-VMC during variable-velocity movement remain largely confined within a consistent range, and the curve demonstrates a high degree of symmetry. This indicates that the BP-VMC sustains a stable motion state through adaptive parameter adjustments, while also ensuring a high level of consistency in the motion control across the four legs.

In order to facilitate a more intuitive comparison of the motion stability between VMC and BP-VMC, we calculated the absolute deviations of each attitude angle as well as the instability indices, based on the relevant formulas (39) and (40) (refer to Figure 32 and Table III). The comparative analysis indicates that the mean standard deviation of BP-VMC across the three primary attitude angles is less than that of VMC. Especially in the roll and pitch angles, the standard deviation of BP-VMC is 74.18% and 60.88% of VMC, respectively, indicating a smaller angle deviation and a more stable attitude. The mean value of the instability metric for VMC is 2.278×10^{-2}, while the mean value for BP-VMC is 1.500×10^{-2}, representing a 34.15% improvement. Additionally, both controllers' maximum instability indices occur during the transition between high and low velocities. The maximum instability metric for VMC reaches 9.488×10^{-2}, while BP-VMC only reaches 5.595×10^{-2}. This suggests that under varying velocity conditions, BP-VMC can maintain good control stability and suppress significant changes in

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

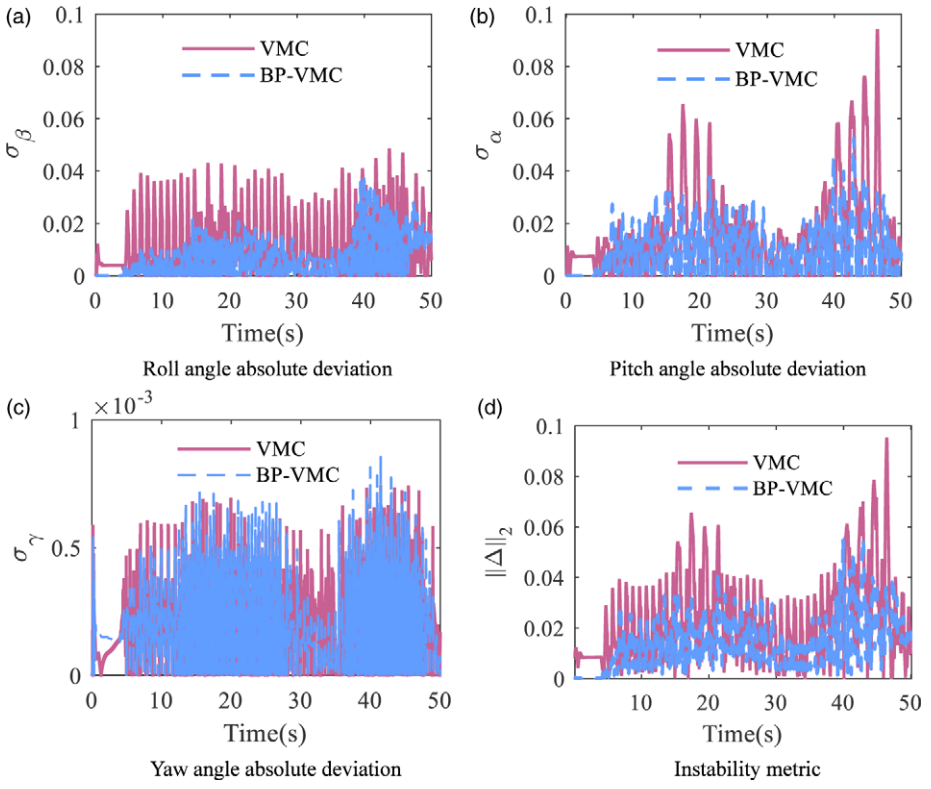


Figure 32. *Attitude angle absolute deviation and instability metric.*

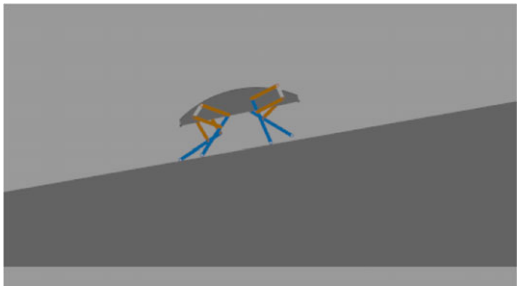


Figure 33. *10° slope.*

the body's attitude angle to a certain extent. Overall, whether analyzing the entire simulation period or any specific time point, the stability of the BP-VMC controller is superior to VMC.

To facilitate the subsequent research on control in complex terrain environments, we further verify the stability of BP-VMC under sloped terrain conditions. The slope gradient is set to 10°, and the walking velocity is 0.4 m/s, as shown in Figures 33 and 34.

The curve results in Figures 35–36 are listed in Table IV. Under sloped conditions, the instability metrics for VMC and BP-VMC are 4.140×10^{-2} and 3.063×10^{-2}, respectively, with BP-VMC showing a 26.01% improvement in stability. Therefore, using the BP-VMC control strategy in complex environmental conditions will provide the quadruped robot with better stability.

6.5 Variable stride frequency (VSF) working condition

In previous simulation experiments, the robot worked under conditions where the stride frequency remained constant and the stride length varied according to the tracking velocity. To further verify the

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

Table IV. *Range of attitude angle fluctuations and instability metric mean value.*

	Roll angle α	**Pitch angle β**	**Yaw angle γ**	**Instability metric $\|\Delta\|_2$**
VMC	$-0.057 \sim 0.028$ rad	$-0.222 \sim$ -0.172 rad	$-7 \times 10^{-4} \sim 7 \times 10^{-4}$ rad	4.140×10^{-2}
BP-VMC	$-0.035 \sim 0.044$ rad	$-0.198 \sim$ -0.165 rad	$-8 \times 10^{-4} \sim 8 \times 10^{-4}$ rad	3.063×10^{-2}

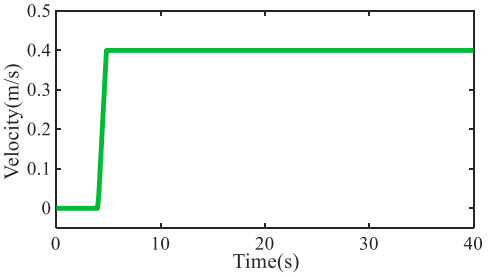


Figure 34. *Walking velocity curve.*

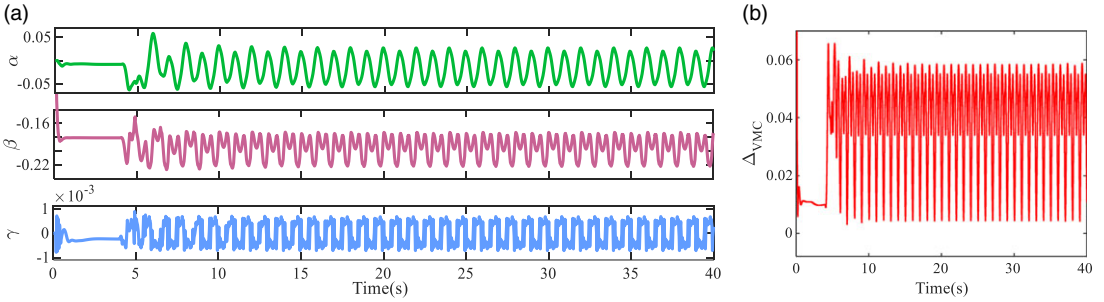


Figure 35. *VMC attitude angle and instability metric.*

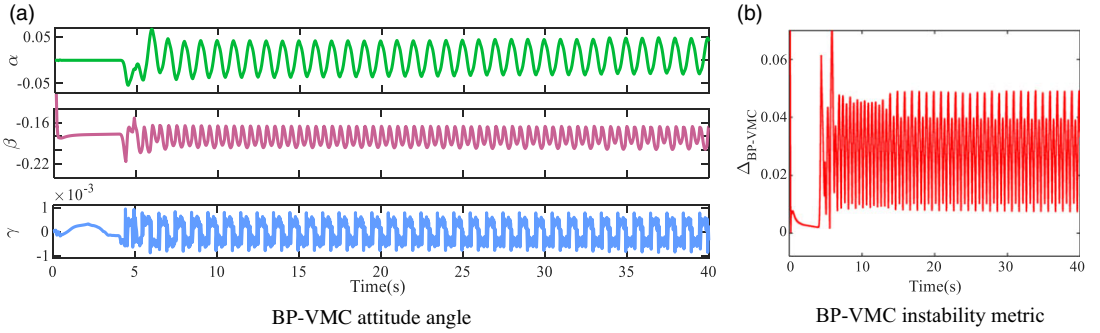


Figure 36. *BP-VMC attitude angle and instability metric.*

performance optimization effect of the BP-VMC controller, additional simulations were conducted under conditions where the stride length remained constant and the stride frequency varied. The constant stride length was set at $S = 0.15$ m, and the variation curve of the motion period T is shown in Figure 37, which forms the desired foot trajectory as depicted in Figure 38. Other parameters remain consistent with those in Table I.

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

Table V. *Relevant error data under VSF working condition.*

	VMC	**BP-VMC**
RMSE in the X direction (m)	1.245×10^{-2}	0.966×10^{-2}
RMSE in the Z direction (m)	8.701×10^{-3}	3.746×10^{-3}
Mean tracking error of the foot trajectory (m)	1.384×10^{-2}	0.933×10^{-2}
Mean absolute deviation of roll angle (rad)	1.692×10^{-2}	1.596×10^{-2}
Mean absolute deviation of pitch angle (rad)	1.259×10^{-2}	0.798×10^{-2}
Mean absolute deviation of yaw angle (rad)	2.005×10^{-4}	2.137×10^{-4}
Mean value of instability metric	2.353×10^{-2}	1.872×10^{-2}

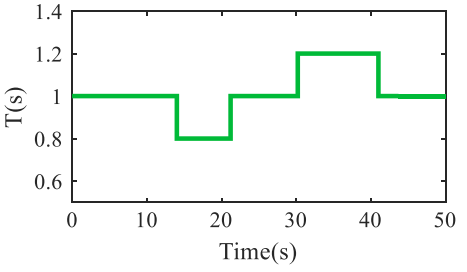


Figure 37. *Motion cycle curve.*

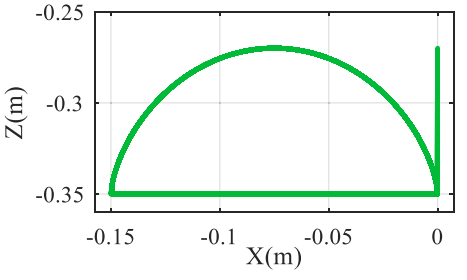


Figure 38. *Desired foot trajectory.*

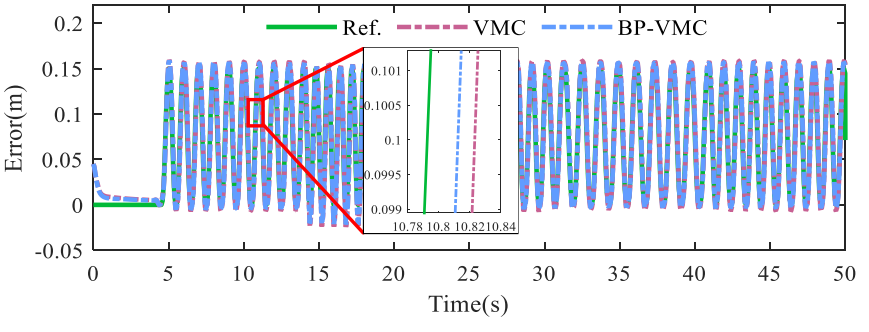


Figure 39. *Stride length in the X direction under VSF working condition.*

Since this working condition mainly involves tracking a fixed stride length under different movement cycle conditions, the analysis focuses primarily on trajectory tracking and motion stability. The resulting curves are presented in Figures 39–44.

The relevant error data for trajectory tracking and stability assessment are presented in Table V. The RMSE in the X direction, the RMSE in the Z direction, and the mean tracking error of the foot trajectory

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

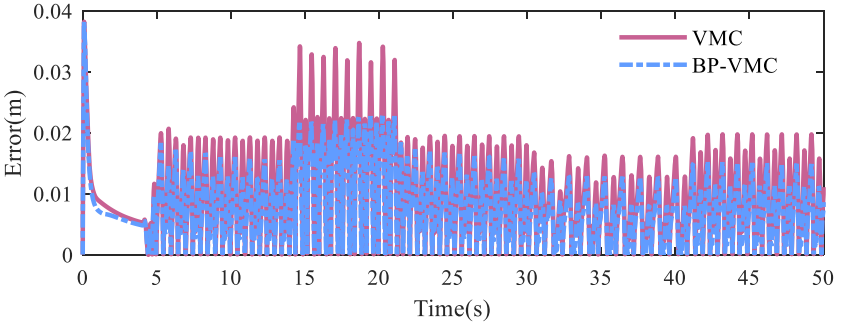


Figure 40. *Absolute error in the X direction under VSF working condition.*

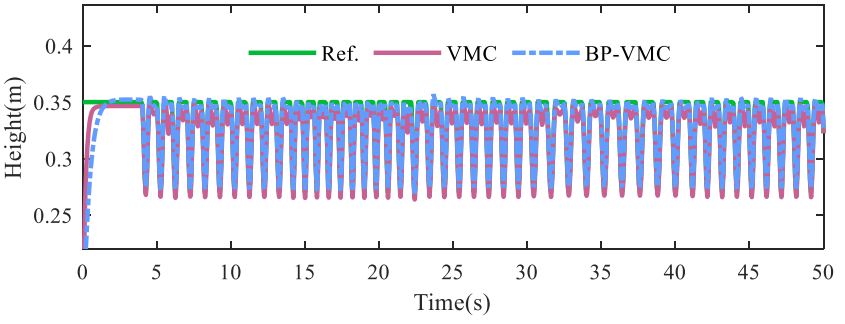


Figure 41. *Height between the foot-end and the center of mass under VSF working condition.*

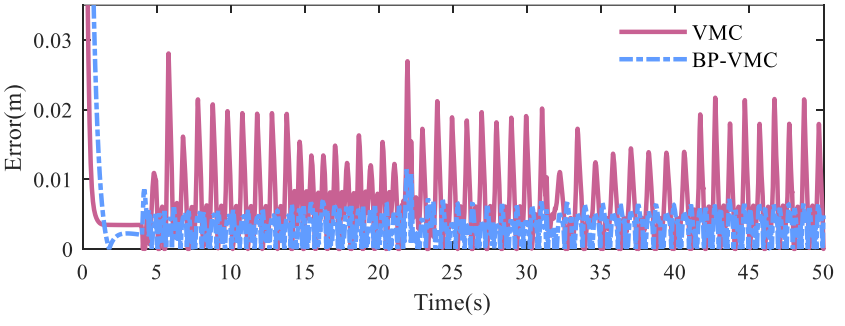


Figure 42. *Absolute error in the Z direction under VSF working condition.*

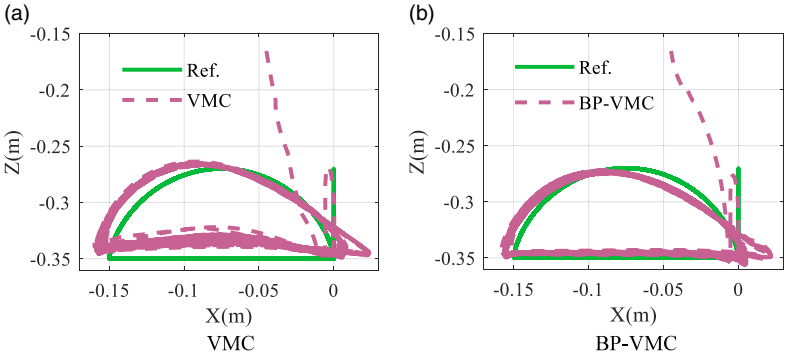


Figure 43. *Foot-end movement trajectory under VSF working condition.*

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

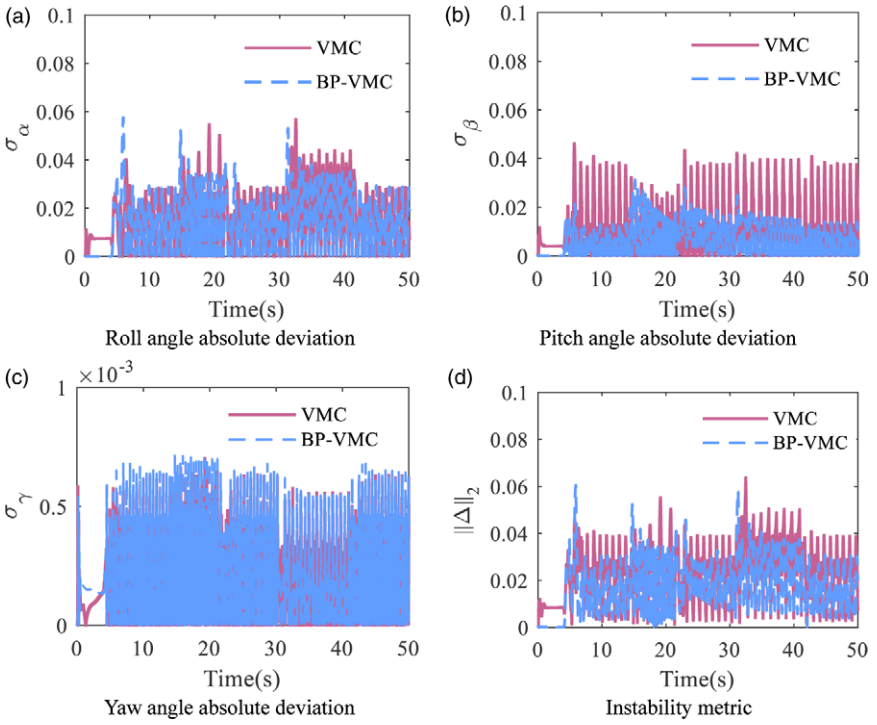


Figure 44. *Attitude angle absolute deviation and instability metric under VSF working condition.*

characterize the accuracy of trajectory tracking. These above values of BP-VMC are 77.59%, 43.05%, and 67.41% of VMC, respectively, suggesting that BP-VMC achieves enhanced control accuracy and reduced tracking errors. Furthermore, the mean absolute deviations of the roll angle, pitch angle, and yaw angle, along with the mean value of the instability metric, characterize the motion stability throughout the control process. BP-VMC exhibits comparable mean absolute errors in the roll and yaw angles relative to VMC; however, the mean absolute error in the pitch angle and the mean value of the instability metric for BP-VMC are 63.38% and 79.56% of those for VMC, respectively. Consequently, BP-VMC demonstrates superior control stability.

6.6 Comparison between BPNN optimization and fuzzy optimization

Several researchers have suggested the application of fuzzy algorithms to enhance VMC by adaptively modifying the virtual parameter matrices K and B [32, 40]. In light of this, a Fuzzy VMC controller was developed, as referenced in the literature [40], to evaluate the control performance of BP-VMC in comparison to Fuzzy VMC under equivalent control conditions.

The performance of the BP-VMC and Fuzzy VMC controllers is compared from the aspects of stride length tracking error, velocity tracking error, foot trajectory tracking error, and motion stability, resulting in the curves shown in Figures 45–48. And these comparisons are also summarized in Table VI.

From the simulation results, both BP-VMC and Fuzzy VMC show significant improvements in control performance compared to traditional VMC. BP-VMC outperforms Fuzzy VMC in terms of control accuracy and velocity tracking accuracy, while Fuzzy VMC is superior to BP-VMC in terms of motion control stability.

The optimization principle of BPNN has been discussed in Section 4.1, and the principle of fuzzy optimization is illustrated in Figure 49.

From the perspective of optimization principles, fuzzy algorithms are based on experience, where fuzzy rule tables are artificially formulated according to the characteristics of the system to correct

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

Table VI. *Relevant error data under VSF working condition.*

	VMC	Fuzzy VMC	BP-VMC
RMSE in the X direction (m)	1.783×10^{-2}	1.591×10^{-2}	1.310×10^{-2}
Mean tracking error of the foot trajectory (m)	1.949×10^{-2}	1.751×10^{-2}	1.402×10^{-2}
RMSE for velocity tracking (m/s)	7.456×10^{-2}	6.975×10^{-2}	4.227×10^{-2}
Mean value of instability metric	3.487×10^{-2}	1.807×10^{-2}	2.184×10^{-2}

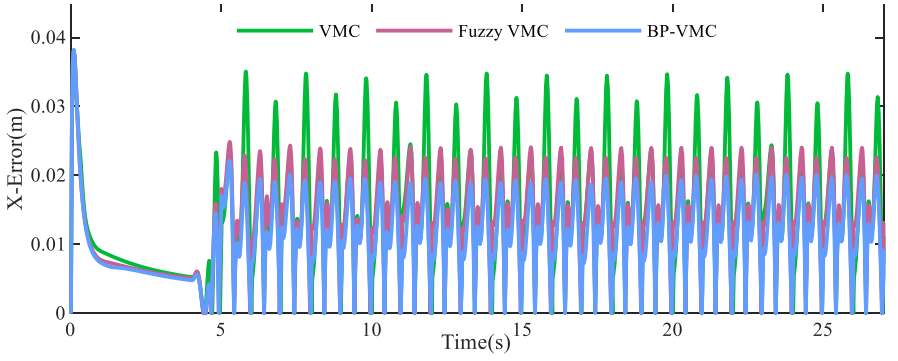


Figure 45. *Tracking error of the stride length in the X direction for different controllers.*

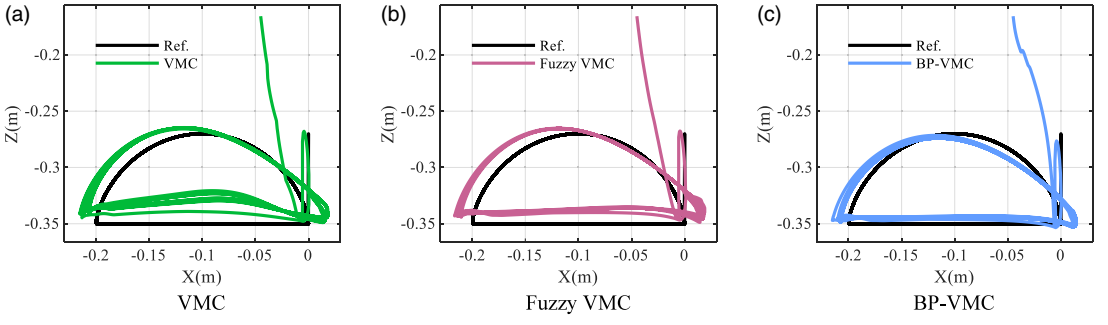


Figure 46. *Foot-end movement trajectories of different controllers.*

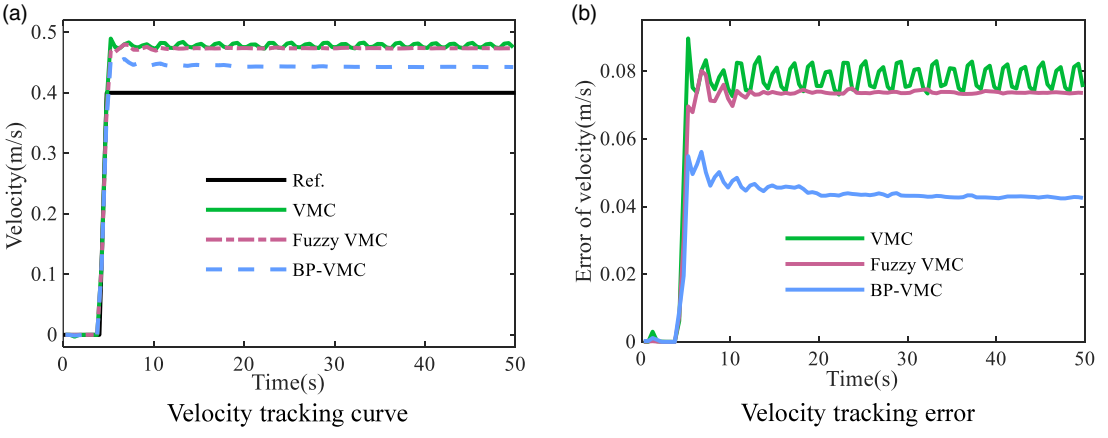


Figure 47. *Velocity tracking results of different controllers.*

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

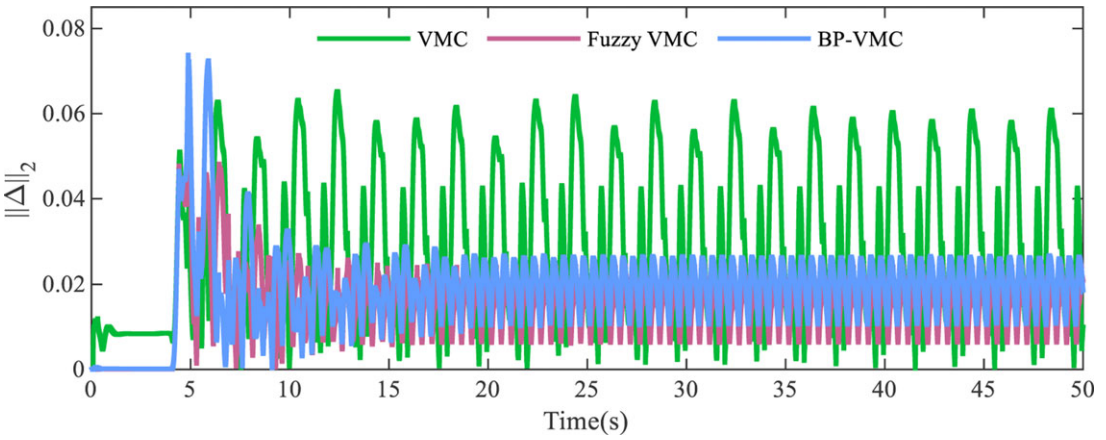


Figure 48. *Instability metrics of different controllers.*

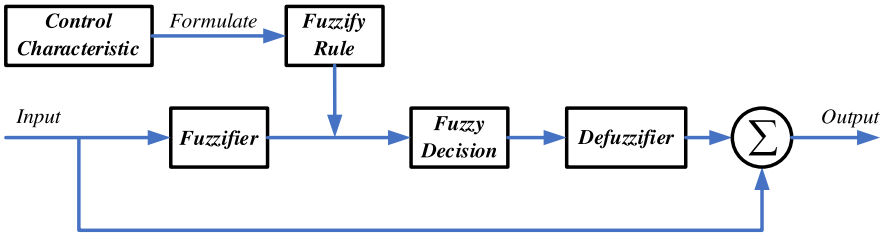


Figure 49. *Fuzzy optimization principle block diagram.*

and improve parameters. The rationality of rule formulation has a decisive impact on the final control performance [41, 42]. However, it is difficult to qualitatively determine the rationality of fuzzy rule formulation. And during use, fuzzy rules are also hard to adjust autonomously. The results obtained are merely the optimal results under the established rule conditions, which can easily lead to local optimal issues.

BPNN builds a neural network structure and autonomously learns and trains based on sample data, making adaptive adjustments to parameters according to error conditions [43–45]. During the learning and training process of the BPNN algorithm, the core weight matrix is continuously optimized and adjusted, while also introducing inertia factors and learning rates, which can effectively avoid local optimal problems and achieve global optimal results.

Therefore, from the analysis of optimization principles, BP-VMC can achieve global optimal control better than Fuzzy VMC, and it also has broader applicability.

7. Conclusion

This paper focuses on the motion control necessities of quadruped robots and employs an intuitive and effective control framework known as VMC, to simplify the dynamics modeling process of quadruped robots. To address the insufficient dynamic performance and control accuracy caused by the simplified dynamics in VMC, an adaptive VMC method optimized by a BPNN is proposed. This approach facilitates the adaptive modification of the virtual component model, thereby compensating for modeling inaccuracies. Additionally, an error integral compensator (IC) is introduced into the height controller in the Z direction, with the BPNN autonomously calibrating the integral coefficient to achieve compensation for system inertia and external errors, further improving control accuracy.

In the simulation experiments, the parameter values of VMC were adaptively tuned by BPNN. The simulation results of VMC showed that the tuned parameters achieved good control performance,

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

demonstrating the reliability of BPNN parameter tuning. Subsequently, the control performance of BP-VMC was analyzed with VMC as the control group. The experimental results showed that under the conditions of fixed stride frequency and varying stride length, the BP-VMC method optimized the errors in trajectory tracking and velocity tracking by 32.73% and 44.57%, respectively, compared to VMC, while also improving the instability metric by 34.15%. Under the conditions of varying stride frequency and fixed stride length, BP-VMC optimized the trajectory tracking and instability metric by 32.59% and 20.44%, respectively, compared to VMC, verifying the optimized effects of BP-VMC in terms of accuracy in trajectory tracking, velocity tracking, and stability of motion control under different working conditions. Simultaneously, simulation comparison experiments between BP-VMC and Fuzzy VMC were conducted. Through a comparative analysis of the simulation results and optimization principles, the findings indicate that BP-VMC exhibits superior control accuracy and achieves better global optimization compared to Fuzzy VMC.

Through the research presented in this paper, the VMC method has been optimized and improved, enhancing its control adaptability, stability, and accuracy. This serves as an exploration and expansion of the motion control methods for quadruped robots in complex environments, further promoting the application of quadruped robots in military, rescue, agricultural, and other fields. Further research on the motion control of BP-VMC under complex terrain conditions will be conducted in the future.

Author contributions. Liu Jianwen conceived and designed the study. Xu Xiaojun and Wang Wenhao provided relevant theoretical guidance and financial support. Liu Jianwen, Tang Yuanjiang, and Lu Shengyang carried out relevant simulation experiments. Liu Jianwen primarily wrote the article, and Xu Xiaojun, Wang Wenhao, Tang Yuanjiang, and Lu Shengyang revised and polished it.

Financial support. This work was financially supported by the National Natural Science Foundation of China Youth Fund (grant no. 52202485) and the National University of Defense Technology Research Program (grant no. ZK24-22).

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] M. M. Gor, P. M. Pathak, A. K. Samantaray, J. M. Yang and S.W. Kwak, "Control oriented model-based simulation and experimental studies on a compliant legged quadruped robot," *Robot. Auton. Syst.* **72**(C), 217–234 (2015).

[2] T. Hamid and M. Nasser, "A study on quadruped mobile robots," *Mech. Mach. Theory* **190**, 105448 (2023).

[3] C. Jiandong, Z. Jinzhu, T. Wang, M. Jiahao, L. Senlin and L. Miao, "Mechanism design and dynamic switching modal control of the wheel-legged separation quadruped robot," *Robotica* **42**(3), 660–683 (2024).

[4] S. Kousik, S. Rahul, A. R. Annigeri, U. B. Praveen Kumar and M. N. Bharath, "Performance analysis of quadruped robot designed with Desai's walking leg mechanism," *Aust. J. Mech. Eng.* **22**(2), 282–295 (2024).

[5] I. Amir, G. Yuan and G. Yan, "Provably stabilizing controllers for quadrupedal robot locomotion on dynamic rigid platforms," *IEEE/ASME Trans. Mechatron.* **25**(4), 2035–2044 (2020).

[6] T. De Paula Daniel, G. E. Paciencia and B.-V. Mauricio, "Dynamic modeling and simulation of a torque-controlled spatial quadruped robot," *Robotica* **42**(8), 1–20 (2024).

[7] Z. Huaizhi, Z. Junhui, L. Jiang, Z. Kun, S. Jun, L. Zhenyu, W. Ke, W. Yanli and X. Bing, "Bionic lightweight design of limb leg units for hydraulic quadruped robots by additive manufacturing and topology optimization," *Bio-Des. Manuf.* **7**(1), 1–13 (2024).

[8] S. Akshit, M. Vijay, V. Krishnan, M. Sandeep and M. D. Arun, "Formation control and trajectory tracking of nonholonomic mobile robots," *IEEE Trans. Control Syst. Trans.* **26**(6), 2250–2258 (2018).

[9] Y. A. Fan, Z. C. Pei, C. Wang, M. Li, Z. Y. Tang and Q. H. Liu, "A review of quadruped robots: Structure, control, and autonomous motion," *Adv. Intell. Syst.* **6**(6) (2024).

[10] C. Hui, L. Yibin, R. Song, Z. Guoteng, Z. Qin, L. Song, H. Jinmian, X. Yaxian, Y. Ming, Z. Guoxuan and Y. Zhiyuan, "A survey of the development of quadruped robots : Joint configuration, dynamic locomotion control method and mobile manipulation approach," *Biomimetic Intell. Rob.* **1**(1), 9–21 (2022).

[11] L. Shengyang, J. Yue, Z. Lei and X. Xiaojun, "Adaptive differential steering strategy for distributed driving unmanned ground vehicle with variable configurations based on modified localized modelling sliding mode control," *ISA Trans.* **151**, 391–408 (2024).

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

[12] X. Xiaojun, L. Shengyang, Y. Jiang, L. Jianwen and Z. Lei, "Driving strategy of unmanned ground vehicle under split-docking road conditions based on improved EKF and PID-modified SMC," *Adv. Eng. Inf.* **62**(Part C), 102830 (2024).

[13] H. Takeuchi, "Development of MEL HORSE". **In**: *19th Annual Conference of the Robotics Society of Japan (RSJ2001)*, Tokyo, Japan, September 18-20 (2001).

[14] R. K. K. Yoneda and S. Hirose, "Shigeo Hirose "Feedforward and feedback trot gait control for quadruped walking vehicle," *Auton. Rob.* **12**(2), 157–172 (2002).

[15] H. Raibert Marc and E. Sutherland Ivan, "Machines that walk," *Sci. Am.* **248**(1), 44–53 (1983).

[16] H. Raibert Marc and R. Tello Ernest, "Legged robots that balance," *IEEE Exp.* **1**(4), 89 (1986).

[17] J. Pratt and P. Dilworth, "Virtual Model Control of a Bipedal Walking Robot," **In:** *Proceedings of International Conference on Robotics and Automation*, Albuquerque, NM, USA (1997).

[18] X. Aizhen, C. Teng, X. Rong, Z. Guoteng, L. Yibin and F. Yong, "A robust and compliant framework for legged mobile manipulators using virtual model control and whole-body control," *Rob. Auton. Syst.* **164** (C), 104411 (2023).

[19] F. Yasuhiro, K. Hiroshi and H. Cohen Avis, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *Int. J. Rob. Res.* **22**(3-4), 187–202 (2003).

[20] K. Hiroshi, F. Yasuhiro and H. Cohen Avis, "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts," *Int. J. Rob. Res.* **26**(5), 475–490 (2007).

[21] S. Linlin, W. Wei and Y. Jianqiang, "Active Impact Motion for a Quadruped Robot," **In:** *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, Hong Kong, China (2020).

[22] F. Michele, K. Amina, B. Aubakir, G. Giuseppina, F. Alessio Mauro and B. Matteo, "A neuromorphic control architecture for a biped robot," *Rob. Auton. Syst.* **120**(0), 103244 (2019).

[23] J. Xin, S. Tao and S. Kai, "Design of an Effective Trajectory Control Method for Quadruped Robot ia Neural Network," **In:** *2018 13th World Congress On Intelligent Control and Automation (WCICA), Changsha, China* (2018).

[24] W. Shangyu, L. Xianqing and Y. Linqi, "Balance control of an inverted pendulum on a quadruped robot by reinforcement learning," *J. Phys. Conf. Ser.* **2187**(1), 012024 (2022).

[25] Q. Ji, G. Haibo, H. Su, H. Liangliang, S. Bo, H. Mingying, Y. Haitao and D. Zongquan, "Reinforcement learning-based stable jump control method for asteroid-exploration quadruped robots," *Aerospace Sci. Technol.* **142**(11) (Part B), 108689 (2023).

[26] G. Sibo, L. Shangke, Z. Hongyin and W. Donglin, "Continual reinforcement learning for quadruped robot locomotion," *Entropy* **26**(1), 93 (2024).

[27] C.-M. Chew and G. A. Pratt, "Dynamic bipedal walking assisted by learning," *Robotica* **20**(5), 477–491 (2002).

[28] B. Gerardo, J. Powell Matthew, B. Katz, D. Carlo Jared, W. M. Patrick and K. Sangbae, "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot," **In:** *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain (2018).

[29] J. Di Carlo, M. Wensing Patrick, B. Katz, B. Gerardo and K. Sangbae, "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control," **In:** *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain (2018).

[30] G. T. Zhang, X. W. Rong, C. Hui, Y. B. Li and B. Li, "Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control," *Adv. Rob.* **30**(4), 284–297 (2016).

[31] T. Chen, X. B. Sun, Z. Xu, Y. B. Li, X. W. Rong and L. L. Zhou, "A trot and flying trot control method for quadruped robot based on optimal foot force distribution," *J. Bionic Eng.* **16**(4), 621–632 (2019).

[32] B. Sabri, "On the implementation of Fuzzy VMC for an under actuated system," *IEEE Access* **7**, 163578–163588 (2019).

[33] Y. Y. X. Li, Z. Xu, Y. Yu, J. Xu and F. Xiao, "Design of trot gait parameters planning system for parallel quadruped robot based on virtual model controller and fuzzy neural network," *ISA Trans.* **157**, 510–529 (2024).

[34] H. X. Xie, M. Ahmadi, J. Z. Shang and Z. R. Luo, "An intuitive approach for quadruped robot trotting based on virtual model control," *Proc. Inst. Mech. Eng. Part I: J. Syst. Control Eng.* **229**(4), 342–355 (2015).

[35] S. V. Bhaskar, R. Manish and G. C. Nandi, "Biometric gait identification based on a multilayer perceptron," *Rob. Auton. Syst.* **65**(1), 65–75 (2015).

[36] H.-N. Robert, "Theory of the backpropagation neural network," *Neural Netw.* **1**(1Suppl), 445 (1988).

[37] I. A. Basheer and M. Hajmeer, "Artificial neural networks: Fundamentals, computing, design, and application.," *J. Microbiol. Methods* **43**(1), 3–31 (2000).

[38] Z. Yue, G. Yue, Q. Sun, T. Yuan, M. Liheng and G. Feng, "A real-time low-computation cost human-following framework in outdoor environment for legged robots," *Rob. Auton. Syst.* **146** (C), 103899 (2021).

[39] C. Guangrong, G. Sheng, H. Bowen and W. Junzheng, "Virtual model control for quadruped robots," *IEEE Access* **8**, 140736–140751 (2020).

[40] S. Y. Y. Zhong and Y. Yulong, "Motion VMC method of quadruped robot based on adaptive fuzzy algorithm," *Appl. Sci. Technol.* **49**(1), 53–58 (2022).

[41] T. T. Deme, G. Manavaalan, K. Halder, V. H. Kumar, P. S. Sundar, J. Nishant, J. F. Orlando Maria and H. Y. Vijay, "Trajectory tracking control of a mobile robot using fuzzy logic controller with optimal parameters," *Robotica* **42**(8), 1–24 (2024).

[42] Y. Ming, D. Haigang, X. Feng, L. Peigang, L. Yanfei and L. Haiyang, "Improved dynamic windows approach based on energy consumption management and fuzzy logic control for local path planning of mobile robots," *Comput. Ind. Eng.* **187**, 109767 (2024).

[43] H. Zhijun, "Simulation of football based on PID controller and BP neural network," *Microprocess. Microsyst.* **81**, 103695 (2021).

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press

[44] Z. Ming-Li, Z. Yi-Jie, H. Xiao-Long and G. Zheng-Jie, "Adaptive PID control and its application based on a double-layer BP neural network," *Processes*, **9**(1475), 1475 (2021).
[45] Z. Yunkang, H. Xiaohui, S. Faming and Z. Xiangpo, "Research on the optimization of the PID control method for an EOD robotic manipulator using the PSO algorithm for BP neural networks," *Actuators* **13**(10), 386 (2024).

Cite this article: J. Liu, X. Xu, W. Wang, Y. Tang and S. Lu (2025). "Adaptive optimization for virtual model control of quadruped robots based on BP neural network", Robotica **43**, 1258–1290. https://doi.org/10.1017/S0263574725000244

https://doi.org/10.1017/S0263574725000244 Published online by Cambridge University Press