Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering) 2023 24(6):516-530
www.jzus.zju.edu.cn; www.springer.com/journal/11582
E-mail: jzus_a@zju.edu.cn

Research Article
https://doi.org/10.1631/jzus.A2200310

A stability locomotion-control strategy for quadruped robots with center-of-mass dynamic planning

Yangyang HAN[1], Guoping LIU[1], Zhenyu LU[1]✉, Huaizhi ZONG[2], Junhui ZHANG[2], Feifei ZHONG[1], Liyu GAO[1]

[1]*School of Advanced Manufacturing, Nanchang University, Nanchang 330031, China*
[2]*State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310058, China*

Abstract: Locomotion stability is essential for controlling quadruped robots and adapting them to unstructured terrain. We propose a control strategy with center-of-mass (CoM) dynamic planning for the stable locomotion of these robots. The motion trajectories of the swing legs are synchronized with the CoM of the robot. To implement the synchronous control scheme, we adjusted the swing legs to form a support triangle. The strategy is applicable to both static walk gait and dynamic trot gait. In the motion control processes of the robot legs, the distribution of the ground reaction forces is optimized to minimize joint torque and locomotion energy consumption. We also used an improved joint-torque controller with varied controller coefficients in the stance and swing phases. The simulation and experimental results demonstrate that the robot can complete omnidirectional locomotion in both walk and trot gaits. At a given locomotion speed, the stability margins for the robot during walking and trotting were 27.25% and 37.25% higher, respectively, than in the scheme without CoM planning. The control strategy with energy consumption optimization (ECO) reduced the energy consumption of the robot in walk and trot gaits by 11.25% and 13.83%, respectively, from those of the control scheme without ECO.

Key words: Center-of-mass (CoM) planning; Quadruped robot; Cooperative scheme; Ground reaction forces; Stability margin

1 Introduction

Quadruped robots have been extensively researched owing to their extreme mobility and adaptability to different terrains; for example, the well-known ANYmal (Hutter et al., 2016, 2017), HyQ (Focchi et al., 2017; Gonzalez et al., 2020), and MIT Cheetah series (Hyun et al., 2014; Park et al., 2017). Stability is the basis for quadruped robots to locomote over complex terrain environments. Inspired by the rhythmic movement of organisms, some researchers have introduced locomotion control with a central pattern generator (CPG) to increase the stability of quadruped robots. Shao et al. (2022) proposed a new control framework based on CPG phase guidance, which can generate various gaits and realize transition between different gaits. Zhang Y et al. (2021) proposed a dynamics-based CPG network control system and simulated the trot gait of a quadruped robot. Fukui et al. (2019) demonstrated that a CPG network with vestibular feedback was an effective and practical gait generator that enabled autonomous gait transitions. Liu and Zhang (2020) established a van der Pol CPG network for a quadruped robot with four primary gaits. They analyzed the existing conditions of model bifurcation and the coupling strength range between the oscillators. However, modulating various CPG models for robot position control is a complex and cumbersome task because the model parameters usually have no apparent physical meaning.

Other researchers have focused on gait planning to improve the stability of robots. In a comparison study of several stable static gaits, Hao et al. (2020) identified the static gait that maximized the stability margin. Wang et al. (2020) attempted to improve the walking velocity of quadruped robots with static gaits. They proposed two methods for walking on an assumed set of irregular footholds. Yeom and Bae (2021) proposed a gait-stabilization algorithm that adjusts the contact time between the robot legs and the ground.

✉ Zhenyu LU, luzhenyu@ncu.edu.cn
Yangyang HAN, https://orcid.org/0000-0003-1080-9498
Zhenyu LU, https://orcid.org/0000-0001-7784-5149

Received June 14, 2022; Revision accepted Nov. 28, 2022; Crosschecked Apr. 24, 2023

They demonstrated the stability of their algorithm by using a Lyapunov function. Chignoli and Wensing (2020) presented a control strategy for quadruped balancing that enabled postural stability in underactuated contact configurations. Zhang et al. (2019) proposed a continuous static gait planning method for robots walking blindly over unknown rough terrain. McClain and Meek (2018) selected an appropriate gait for a quadruped robot assisting human walking by estimating the optimal parameters of stable static gaits. However, locomotion stability of robots has been much more extensively researched for static than for dynamic gaits. In our proposed stability-control strategy, control of static gaits is extended to control of dynamic gaits, thereby expanding the application scope of the stability-margin method.

Some researchers have applied optimization algorithms for torque control of quadruped robots. Using a Hermite interpolation method, Tian et al. (2019) generated a smooth gait under boundary constraints, and established a control framework for quadruped robots based on virtual model control. Zhang ML et al. (2021) proposed a double-layer back-propagation neural network that learns the parameter rules of a proportional-integral-derivative controller. This method avoids the inconvenient adjustment of artificial parameters otherwise necessary for the hydraulic drive unit of a legged robot. di Carlo et al. (2018) converted dynamics locomotion control into a convex optimization problem based on a simplified model. They determined the ground reaction forces (GRFs) for a quadruped robot using model predictive control (MPC). Dudzik et al. (2020) combined regular predictive control and whole-body impulse control into an integrated system for small robots. Several groups have tried various algorithms to optimize function. Pepe et al. (2021) determined the optimal trajectory of a quadruped robot using a genetic algorithm that seeks the optimal gait sequence at different velocities. Srinivas et al. (2021) optimized the joint positions of a robot with eight degrees of freedom (DoFs) using the particle-swarm optimization algorithm. Lee and An (2021) proposed a self-balancing control algorithm for quadruped robots, based on reinforcement learning and an artificial neural network, which replaces the analysis-based control algorithm. Arena et al. (2021) proposed a data-driven neural network based on nonlinear MPC, and compared the simulation results for nonlinear MPC with those for linear MPC. Ding et al. (2021) proposed a representation-free MPC framework that controls various locomotion patterns of robots in 3D space. However, the complex algorithm burdened the computing power of the robot, thereby increasing the difficulty of implementation and the investment cost. Our proposed control strategy, which involves GRF optimization based on MPC (di Carlo et al., 2018), designs the objective function to minimize GRFs and energy consumption and simplifies the optimization algorithm to reduce the amount of computation. Moreover, it has flexible weight coefficients, which allow for different stances and swing phases of the legs. We aim to show how this design reduces the complexity of the control system.

The main contributions of this study are summarized below.

(1) The center-of-mass (CoM) and swing legs of the robot are synchronized to optimize coordination during locomotion. The scheme allows real-time trajectory planning for the CoM, and the stability margin can be flexibly and conveniently adjusted to an appropriate value. The plan can be dynamically expanded to trot gait based on walk gait.

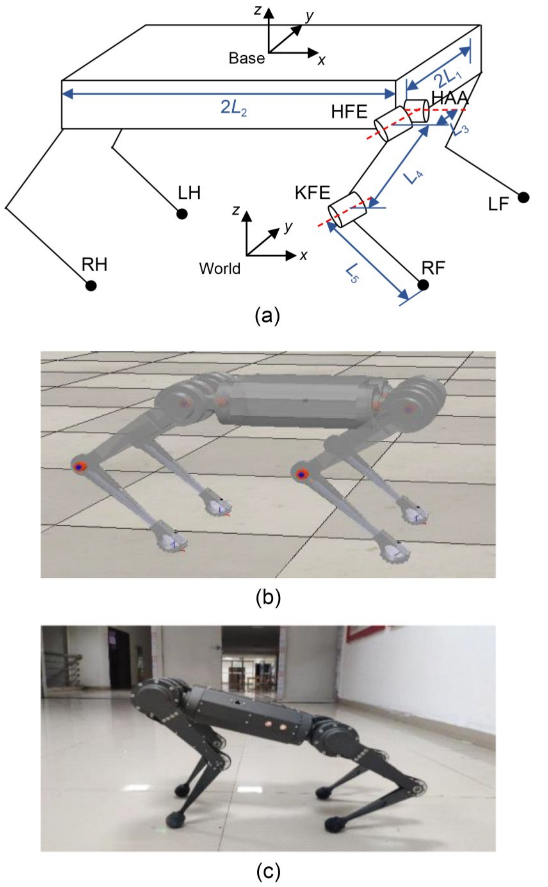
(2) Locomotion stability and energy consumption are simultaneously considered with a constant desired velocity. The stability goal is accomplished by real-time dynamic planning for the CoM. Energy consumption is minimized through the objective function of the quadratic planning of the GRFs. As the forces on the robot legs differ between the stance and swing phases, they are handled under different control strategies. The joint torques of the standing legs are primarily generated by mapping the optimal GRFs, whereas a joint proportional-derivative (PD) controller commands the swinging legs.

(3) The implementation of the designed control strategy of the robot was verified in various ways. Controller performance was demonstrated through simulations and experiments. The tracking errors of the base orientation were analyzed at different velocities. Taking the stability measurement criterion (Gonzalez de Santos et al., 1998) as the judgment measure, we compared the real-time stability margins of the control schemes with and without CoM planning in walk and trot gaits. We also compared the instantaneous power and mean GRFs (in the z direction) of the robot legs with the control strategies, both with and without energy consumption optimization (ECO), in the two gaits; we took the energy model used by Zhou et al. (2021) as the judgment measure.

The remainder of the paper is organized as follows. Section 2 introduces the robot model and the overall design framework of the controller. Section 3 describes each part of the controller and elaborates on the dynamic planning algorithm of the CoM and the principle of the state machine. Section 4 covers the simulation and experimental platforms and the parameter settings of the controller, as well as the simulation and experimental results. Section 5 concludes the study and discusses future work directions.

2 Robot model and control architecture

The employed robot models are shown in Fig. 1, and the physical parameters of the robot are shown in Table 1.



(a)

(b)

(c)

Fig. 1 Robot models: (a) simplified model and coordinate systems; (b) simulation model; (c) experimental prototype. LF, RF, RH, and LH represent four legs (L: left; R: right; F: front; H: hind). HAA: hip-abduction/adduction joint; HFE: hip-flexion/extension joint; KFE: knee-flexion/extension joint

Table 1 Physical robot parameters

Parameter	Value
Mass, m (kg)	10.5
Degree of freedom, DoF	12
Length 1, L_1 (m)	0.049
Length 2, L_2 (m)	0.190
Length 3, L_3 (m)	0.062
Length 4, L_4 (m)	0.209
Length 5, L_5 (m)	0.195

The block diagram of the control system is shown in Fig. 2. The parameter with subscript ‘d’ denotes the desired value of the parameter. Depending on the input parameters, the planner designs the gait sequence, the desired CoM values, and the trajectory of the swing leg. These desired values are used to optimize the distribution of GRFs $\boldsymbol{f}_\mathrm{d}$ based on the dynamic equations. The feedforward torques $\boldsymbol{\tau}_\mathrm{ff}$ are computed by mapping the GRFs to the joint space. The joint position $\boldsymbol{q}_{i,\mathrm{d}}$ and velocity $\dot{\boldsymbol{q}}_{i,\mathrm{d}}$ are calculated through inverse kinematics (IK) of the desired footstep location $\boldsymbol{p}_{i,\mathrm{d}}$ of each swing leg generated by the planner. The desired joint torques $\boldsymbol{\tau}_\mathrm{d}$ of the quadruped robot are obtained by summing the feedback terms $\boldsymbol{\tau}_\mathrm{fb}$ computed by the joint PD controller and the feedforward terms $\boldsymbol{\tau}_\mathrm{ff}$. The robot is equipped with an inertial measurement unit and several joint motor encoders. The sensor data are used for measuring the joint position $\boldsymbol{q}_i$ and velocities $\dot{\boldsymbol{q}}_i$ of each leg, the linear acceleration $\boldsymbol{a}_\mathrm{com}$, and the angular velocity $\boldsymbol{\omega}_\mathrm{B}$ of the base (di Carlo et al., 2018). The current body state of the robot is estimated with a leg-odometry algorithm. The state values are then transmitted to the planner and optimizer for real-time data updates.

3 Locomotion-control strategy

First, we planned the gait and swing-leg trajectories. We then compared two coordination schemes when used for the CoM and legs, and designed an algorithm to implement the dynamic CoM planning. We will also present the state transition process.

3.1 Gait planning

Each typical gait was assigned a corresponding gait pattern (Hyun et al., 2014). The gait sequence and phase relation of walking are shown in Figs. 3a and 3b, respectively. The swing sequence was LF-RH-RF-LH,

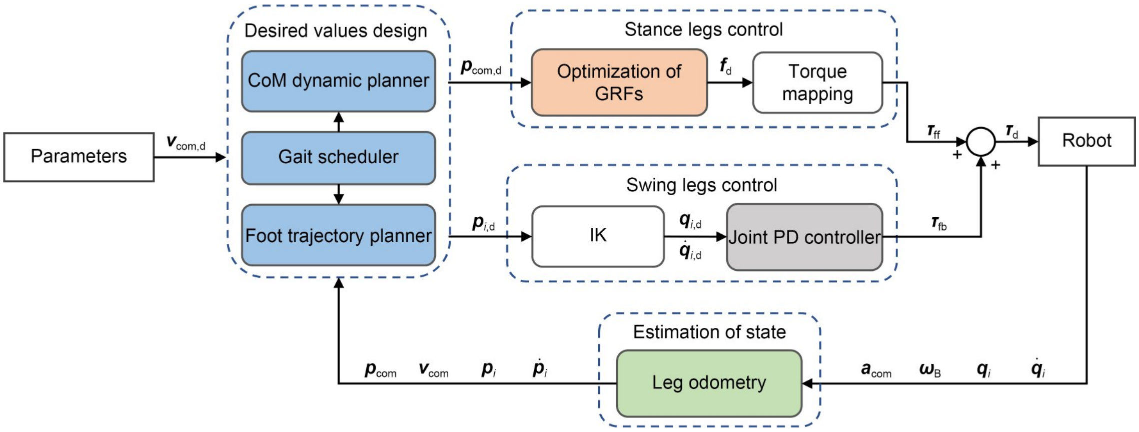

Desired values design
CoM dynamic planner
Gait scheduler
Foot trajectory planner
Parameters
$\boldsymbol{v}_{\text{com,d}}$
$\boldsymbol{p}_{\text{com,d}}$
Stance legs control
Optimization of GRFs
$\boldsymbol{f}_{\text{d}}$
Torque mapping
$\boldsymbol{\tau}_{\text{ff}}$
$\boldsymbol{\tau}_{\text{d}}$
+
+
Robot
Swing legs control
$\boldsymbol{p}_{i,\text{d}}$
IK
$\boldsymbol{q}_{i,\text{d}}$
$\dot{\boldsymbol{q}}_{i,\text{d}}$
Joint PD controller
$\boldsymbol{\tau}_{\text{fb}}$
Estimation of state
Leg odometry
$\boldsymbol{p}_{\text{com}}$ $\boldsymbol{v}_{\text{com}}$ $\boldsymbol{p}_i$ $\dot{\boldsymbol{p}}_i$
$\boldsymbol{a}_{\text{com}}$ $\boldsymbol{\omega}_{\text{B}}$ $\boldsymbol{q}_i$ $\dot{\boldsymbol{q}}_i$

Fig. 2 Block diagram of the control system. $\boldsymbol{v}_{\text{com}}$ is the velocity of the CoM; $\boldsymbol{p}_{\text{com}}$ is the position of the CoM; $\boldsymbol{p}_i$ is the footstep location

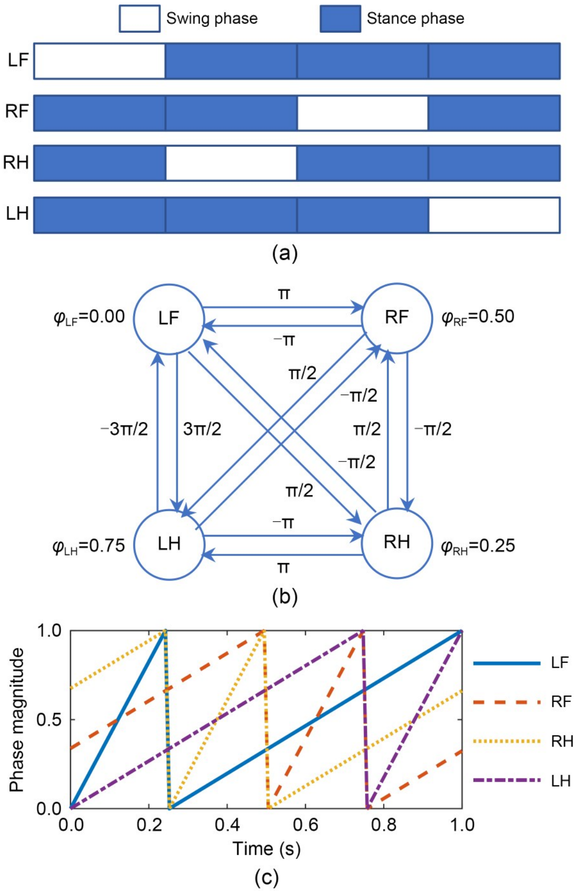


Fig. 3 Gait pattern for walk gait: (a) gait sequence; (b) phase relation; (c) phase-control signal

and the phase (φ) difference between adjacent swing legs was 0.25. The current state S_i of the ith leg was defined as S_i=0 during the swing phase and S_i=1 during the stance phase. The robot legs were periodically switched between the stance and swing phases depending on the gait type. The phase-control signal (time cycle T=1 s) generated by the gait scheduler for walk gait is shown in Fig. 3c.

3.2 Swing-trajectory planning

The desired velocity of the robot is expressed as

$$\boldsymbol{v}_{\text{com,d}}=[\,v_{x,\text{d}}\quad v_{y,\text{d}}\quad v_{z,\text{d}}\quad \omega_{x,\text{d}}\quad \omega_{y,\text{d}}\quad \omega_{z,\text{d}}\,]^{\text{T}},\qquad(1)$$

where $v_{x,\text{d}}$, $v_{y,\text{d}}$, $v_{z,\text{d}}\in\mathbb{R}$ are linear velocity terms and $\omega_{x,\text{d}}$, $\omega_{y,\text{d}}$, $\omega_{z,\text{d}}\in\mathbb{R}$ are angular velocity terms. The subscript d denotes that these velocity terms are the desired values. The linear velocity $v_{z,\text{d}}$ and angular velocities $\omega_{x,\text{d}}$ and $\omega_{y,\text{d}}$ were set to 0 for smooth movement of the robot.

The desired footstep locations of the ith swing leg are given by

$$\boldsymbol{p}_{i,\text{d}}=\boldsymbol{p}_{i,\text{r}}+\boldsymbol{v}_{\text{com,d}}\cdot\frac{\Delta t}{2}+k\left(\boldsymbol{v}_{\text{com}}-\boldsymbol{v}_{\text{com,d}}\right),\qquad(2)$$

where $\boldsymbol{p}_{i,\text{r}}\in\mathbb{R}^3$ is defined as the position of the ith leg relative to its shoulder, Δt is the stance duration, and $k\in\mathbb{R}$ is the tracking error coefficient (k=0.1 here). The second term on the right of Eq. (2) is the heuristic (Raibert, 1986), and the third term compensates for the tracking error in the CoM velocity.

The swing trajectory is described by a cubic spline, which is generated from the initial position to the desired footstep location. The height of the midpoint position of the trajectory reaches the step height $H_{\text{sw}}\in\mathbb{R}$. The initial and final velocities were set to 0 to reduce the robot's ground-contact impact force.

3.3 Dynamic planning for the CoM

The CoM was dynamically planned in order to enhance its locomotion stability. To coordinate the motions of the CoM with the swing, we first determined the cooperative scheme and then designed the motion trajectory of the CoM.

3.3.1 Synchronous cooperation design

The support triangle, which was determined by the cooperative scheme between the motions of the CoM and swing legs, influenced the planning of the CoM and control strategy. The two cooperative schemes for walk gait are shown in Fig. 4. Figs. 4a and 4b show the asynchronous and synchronous cooperative schemes, respectively, and Figs. 4c and 4d show their corresponding CoM velocities ($\boldsymbol{v}_{x,\mathrm{d}}$=0.2 cm/s).

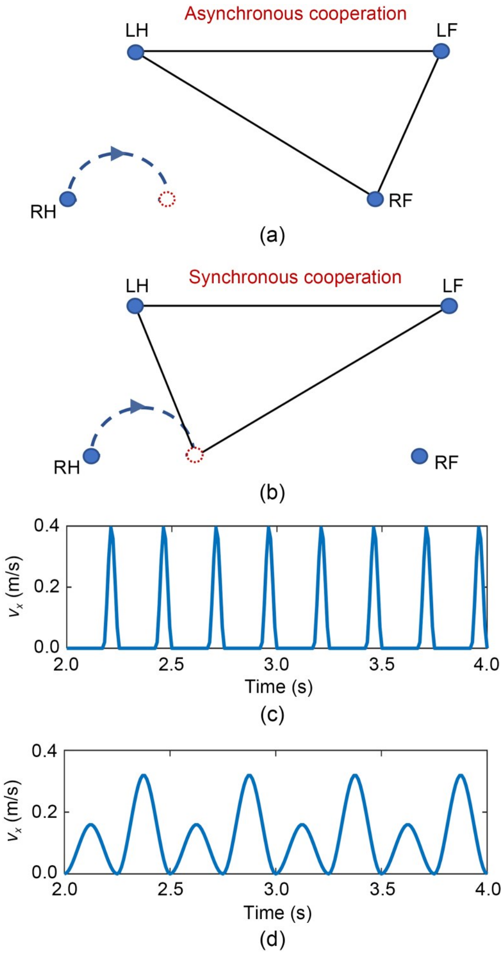


Fig. 4 Two cooperative schemes (walk gait): (a) asynchronous scheme; (b) synchronous scheme; (c) CoM velocity in the asynchronous scheme; (d) CoM velocity in the synchronous scheme. The solid circle represents the current stance position of each leg, the dotted circle represents the planned location of the swing leg, and the areas surrounded by solid black lines represent support triangles in Figs. 4a and 4b

The two schemes generated different support triangles for the same swing leg (RH). In the asynchronous scheme (Fig. 4a), the CoM and swing leg of the robot moved sequentially. The *i*th swing leg moved first, followed by the CoM. The next leg then swung in sequence. In this scheme, the three stance legs formed the support triangle, and the support vertices were the foot positions of the stance legs.

In contrast, the designed synchronous scheme led to simultaneous motions of the CoM and swing leg (Fig.4b). The next leg and the CoM then moved at the same time. In our scheme, the legs other than the next swing leg formed the support triangle. The swing leg became a support leg, and its vertex was the planned footstep location.

The CoM velocities were discontinuous and varied more widely in asynchronous than in synchronous cooperation (Fig. 4c). During the movement of the swing leg, the CoM was stationary, and its velocity was zero. When the CoM moved, its velocity quickly increased from zero to peak. In contrast, as our scheme synchronized the motions of the CoM and swing leg (Fig. 4d), the velocity was more continuous, smoother, and varied over a smaller range than the velocities obtained in the asynchronous scheme.

3.3.2 Motion trajectory of the CoM design

The initial and target positions of the CoM projection are defined as $\boldsymbol{P}_{xy,\mathrm{d}}(t_0)$ and $\boldsymbol{P}_{xy,\mathrm{d}}(t_\mathrm{m})$, respectively, where t_0 and t_m are the start and end times of the CoM planning, respectively. The positions of the three vertices $\boldsymbol{P}_1(x_1, y_1)$, $\boldsymbol{P}_2(x_2, y_2)$, and $\boldsymbol{P}_3(x_3, y_3)$ on the projection plane of the support triangle are depicted in Fig. 5a.

The midpoint $\boldsymbol{P}_0(x_0, y_0)$ of the maximum edge $\overrightarrow{\boldsymbol{P}_1\boldsymbol{P}_2}$ is given by

$$\boldsymbol{P}_0(x_0,\ y_0)=\begin{bmatrix}\frac{1}{2}(x_1+x_2) & \frac{1}{2}(y_1+y_2)\end{bmatrix}^\mathrm{T}. \tag{3}$$

The stability margin d, defining the distance from $\boldsymbol{P}_{xy,\mathrm{d}}(t_\mathrm{m})$ to $\boldsymbol{P}_0(x_0, y_0)$ (Gonzalez de Santos et al., 1998), is calculated using Eq. (4). $\boldsymbol{P}_{xy,\mathrm{d}}(t_\mathrm{m})$ can be adjusted by varying the parameter d in the support-triangle plane. The inequation (5), which constrains the generated target position within the plane of the support triangle, must be satisfied.

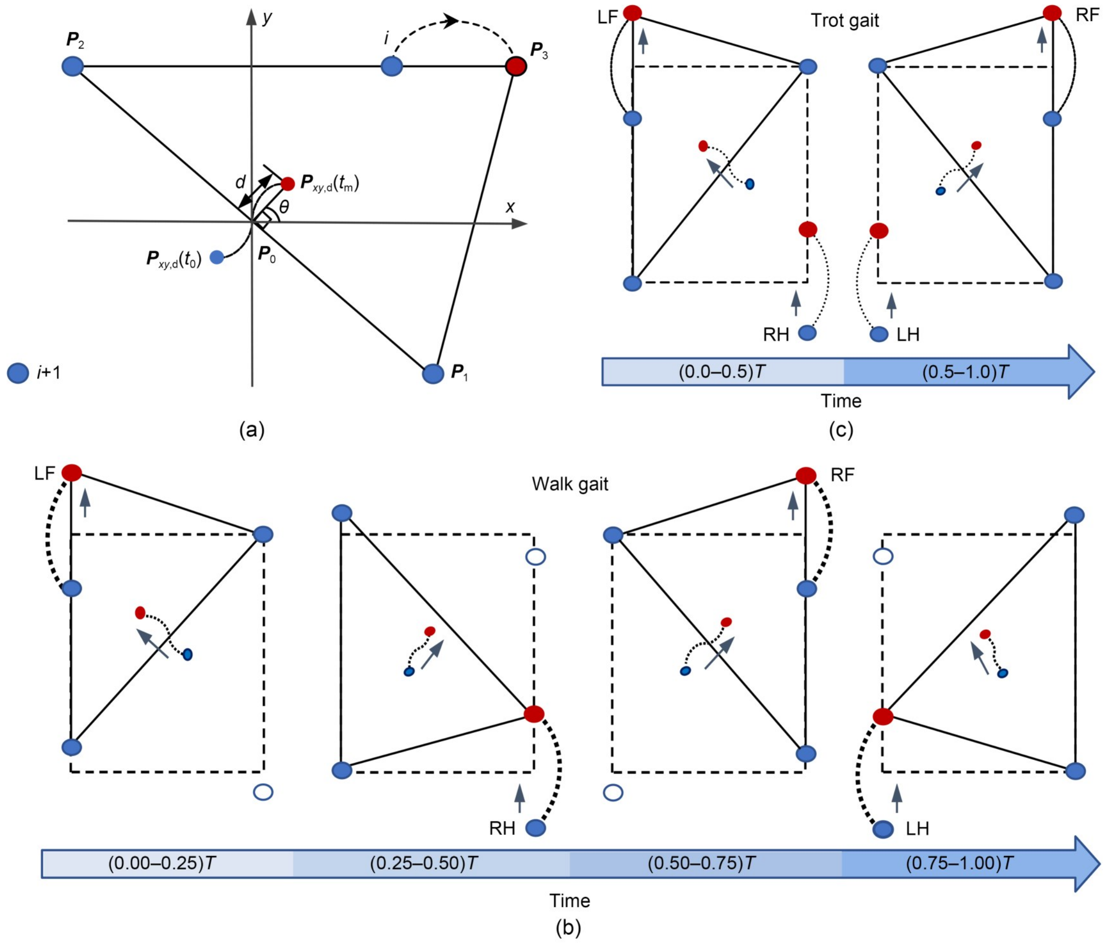


Fig. 5 CoM planning and synchronous cooperative scheme in the projection plane: (a) CoM planning in the projection plane; (b) synchronous cooperation for walk gait; (c) synchronous cooperation for trot gait

$$d=\sqrt{\left(P_{x,\mathrm{d}}(t_\mathrm{m})-x_0\right)^2+\left(P_{y,\mathrm{d}}(t_\mathrm{m})-y_0\right)^2}, \quad (4)$$

$$\sqrt{(x_3-x_0)^2+(y_3-y_0)^2}\geqslant\sqrt{\left(x_3-P_{x,\mathrm{d}}\right)^2+\left(y_3-P_{y,\mathrm{d}}\right)^2}. \quad (5)$$

The angle θ between $\overrightarrow{\boldsymbol{P}_0\boldsymbol{P}_{xy,\mathrm{d}}(t_\mathrm{m})}$ and the x-axis is found by

$$\cos\theta=\frac{P_{x,\mathrm{d}}(t_\mathrm{m})-x_0}{d}. \quad (6)$$

To reduce the movement range of the base on the y-axis, we set the direction to be consistent with the locomotion direction of the robot:

$$\cos\theta=\frac{v_{x,\mathrm{d}}}{\sqrt{\left(v_{x,\mathrm{d}}\right)^2+\left(v_{y,\mathrm{d}}\right)^2}}. \quad (7)$$

A minimum-jerk quintic spline with a fourth-order continuous derivative was planned between the initial and target positions. The CoM velocities and accelerations were set to zero at the start and end times to ensure smooth movement of the CoM along the planned trajectory.

The designed scheme was appropriate for the walking gait, in which one leg swings at a time. Fig. 5b shows the cooperation between the CoM and swing legs of the robot in walking, along with the dynamic trajectory plan of the CoM. Also shown are the implementation details of the synchronous cooperative scheme based on the step sequence generated by the gait scheduler. Note that the vertices of the support triangle are the stance positions in the next phase.

To extend the synchronous cooperative scheme to the trot gait, we took the positions of the stance legs as the two support vertices of the supportive triangle, and the third support vertex employed the planned footstep location of the forward swing leg. Fig. 5c shows the support triangle formed by these vertices and the implementation details of the synchronous cooperative scheme during trotting.

3.3.3 State machine for motion control

The designed robot locomotion was a phased, continuous, and repeated process. Fig. 6 shows the motion states and their transition operations.

When the ith leg is the swing leg, the contact state S_i is set to 0. The desired footstep location $\boldsymbol{p}_{i,\mathrm{d}}(t_{\mathrm{m}})$ is computed by Eq. (2). The cubic spline foot trajectory from the initial position to the target location is determined, and the desired joint position $\boldsymbol{q}_{i,\mathrm{d}}(t)$ is then generated by the IK equations. At this point, the planned footstep location of the ith leg is a vertex of the support triangle. The projection target position $\boldsymbol{p}_{xy,\mathrm{d}}(t_{\mathrm{m}})$ is computed by Eq. (4) depending on the determined support triangle. When the ith leg is the stance leg, the contact state S_i is set to 1. The ith leg is a support vertex of the support triangle (if it is not the next swing leg) and participates in the dynamic planning of the CoM. The CoM moves along the planned quintic spline trajectory from its initial position to the target position. The desired joint position $\boldsymbol{q}_{\mathrm{d}}(t)$ of the stance legs is then generated by the IK equations.

3.4 Motion-state estimation

We used a leg-odometry algorithm (Lin et al., 2005) to estimate the robot's state from the sensor measurements. The state contains the positions and velocities of the feet and the CoM.

In the world coordinate system, the foot position is computed as

$${}^{\mathrm{W}}\boldsymbol{p}_i = {}^{\mathrm{W}}\boldsymbol{p}_{\mathrm{com}} + {}^{\mathrm{W}}\boldsymbol{R}_{\mathrm{B}} \cdot {}^{\mathrm{B}}\boldsymbol{p}_i, \tag{8}$$

where the upper-left superscript denotes the coordinate system (W: world; B: base) of the vector (Fig. 1a), ${}^{\mathrm{W}}\boldsymbol{R}_{\mathrm{B}} \in \mathbb{R}^{3\times3}$ is the rotation matrix from base to world coordinates, and the subscript $i\in\{1, 2, 3, 4\}$ represents the LF, RF, RH, and LH legs, respectively.

The foot position ${}^{\mathrm{B}}\boldsymbol{p}_i$ in the base coordinate is obtained through the forward kinematics by

$$\begin{bmatrix} {}^{\mathrm{B}}p_{i(x)} & {}^{\mathrm{B}}p_{i(y)} & {}^{\mathrm{B}}p_{i(z)} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} L_4 s_2 + L_5 s_{23} + \alpha L_2 \\ \beta(L_3 c_1 + L_1) + L_4 s_1 c_2 + L_5 s_1 c_{23} \\ \beta L_3 s_1 - L_4 c_1 c_2 - L_5 c_1 c_{23} \end{bmatrix}, \tag{9}$$

where c_j=cos($q_{i,j}$), s_j=sin($q_{i,j}$), c_{jk}=cos($q_{i,j}$+$q_{i,k}$), s_{jk}=sin($q_{i,j}$+$q_{i,k}$), $q_{i,j}$ represents the jth joint of the ith leg, and j, $k\in\{1, 2, 3\}$ represent the three joints, HAA, HFE, and KFE, respectively, as defined in Fig. 1a. α, $\beta\in\{-1, 1\}$ are symbolic variables taking the following values:

$$\alpha = \begin{cases} 1, & i=1,\ 4, \\ -1, & i=2,\ 3, \end{cases} \quad \beta = \begin{cases} 1, & i=1,\ 2, \\ -1, & i=3,\ 4. \end{cases} \tag{10}$$

The foot velocity ${}^{\mathrm{B}}\dot{\boldsymbol{p}}_i$ in base coordinates is given by

$${}^{\mathrm{B}}\dot{\boldsymbol{p}}_i = \boldsymbol{J}_i \cdot {}^{\mathrm{B}}\dot{\boldsymbol{q}}_i, \tag{11}$$

where $\boldsymbol{J}_i \in \mathbb{R}^{3\times3}$ is the Jacobian matrix, expressed as

$$\boldsymbol{J}_i = \begin{bmatrix} 0 & L_4 c_2 + L_5 c_{23} & L_5 c_{23} \\ -\beta L_3 s_1 + L_4 c_1 c_2 + L_5 c_1 c_{23} & -L_4 s_1 s_2 + L_5 s_1 s_{23} & -L_5 s_1 s_{23} \\ \beta L_3 c_1 + L_4 s_1 c_2 + L_5 s_1 c_{23} & L_4 c_1 s_2 + L_5 c_1 s_{23} & L_5 c_1 s_{23} \end{bmatrix}. \tag{12}$$

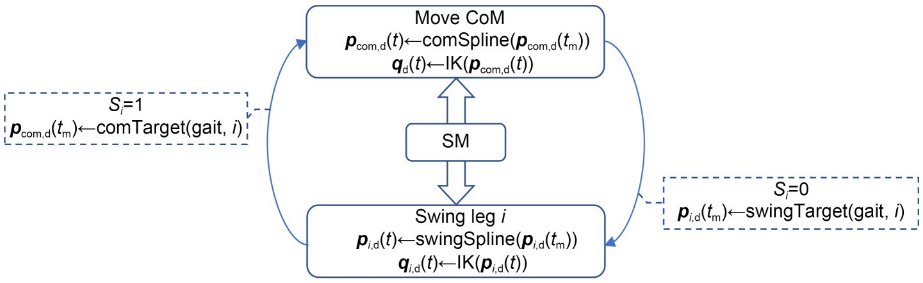

Move CoM
$\boldsymbol{p}_{\mathrm{com,d}}(t)$←comSpline($\boldsymbol{p}_{\mathrm{com,d}}(t_{\mathrm{m}})$)
$\boldsymbol{q}_{\mathrm{d}}(t)$←IK($\boldsymbol{p}_{\mathrm{com,d}}(t)$)
S_i=1
$\boldsymbol{p}_{\mathrm{com,d}}(t_{\mathrm{m}})$←comTarget(gait, i)
SM
S_i=0
$\boldsymbol{p}_{i,\mathrm{d}}(t_{\mathrm{m}})$←swingTarget(gait, i)
Swing leg i
$\boldsymbol{p}_{i,\mathrm{d}}(t)$←swingSpline($\boldsymbol{p}_{i,\mathrm{d}}(t_{\mathrm{m}})$)
$\boldsymbol{q}_{i,\mathrm{d}}(t)$←IK($\boldsymbol{p}_{i,\mathrm{d}}(t)$)

Fig. 6 Logic diagram of the state machine (SM). Rectangles represent the motion state, arrows indicate the direction of state transition, and dotted boxes represent related state-transition operations

Taking the derivative of Eq. (8) and setting ${}^{\mathrm{W}}\dot{\boldsymbol{p}}_i = 0$ to ensure no slippage of the stance legs, the linear velocity of the CoM in world coordinates is given by

$$ {}^{\mathrm{W}}\boldsymbol{v}_{\mathrm{com}} = -{}^{\mathrm{W}}\boldsymbol{R}_{\mathrm{B}}\left({}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{B}\times} \cdot {}^{\mathrm{B}}\boldsymbol{p}_i + {}^{\mathrm{B}}\dot{\boldsymbol{p}}_i\right), \tag{13} $$

where ${}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{B}\times} \in \mathbb{R}^{3\times3}$ is given by

$$ {}^{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{B}\times} = \begin{bmatrix} 0 & -{}^{\mathrm{B}}\omega_z & {}^{\mathrm{B}}\omega_y \\ {}^{\mathrm{B}}\omega_z & 0 & -{}^{\mathrm{B}}\omega_x \\ -{}^{\mathrm{B}}\omega_y & {}^{\mathrm{B}}\omega_x & 0 \end{bmatrix}. $$

The position of the CoM at time t in world coordinates is then determined by

$$ {}^{\mathrm{W}}\boldsymbol{p}_{\mathrm{com}(t)} = {}^{\mathrm{W}}\boldsymbol{p}_{\mathrm{com}(t-\Delta T)} + {}^{\mathrm{W}}\boldsymbol{v}_{\mathrm{com}(t)} \cdot \Delta T, \tag{14} $$

where ΔT is the sampling interval.

In the process of robot-state estimation, the measurements acquired from the sensors include high-frequency noise, which is removed by a first-order low-pass filter.

3.5 Optimization of the GRFs

To optimize the GRFs, we first computed the linear acceleration of the CoM and the angular acceleration of the base, and then solved for the GRFs using the simplified dynamics model.

3.5.1 Dynamics of the simplified rigid body

The desired linear acceleration of CoM and angular acceleration of the robot were designed under the PD control law (Focchi et al., 2017) as

$$ \begin{cases} \boldsymbol{a}_{\mathrm{com,d}} = \boldsymbol{k}_{\mathrm{pa}}\left(\boldsymbol{p}_{\mathrm{com,d}} - \boldsymbol{p}_{\mathrm{com}}\right) + \boldsymbol{k}_{\mathrm{da}}\left(\boldsymbol{v}_{\mathrm{com,d}} - \boldsymbol{v}_{\mathrm{com}}\right), \\ \dot{\boldsymbol{\omega}}_{\mathrm{B,d}} = \boldsymbol{k}_{\mathrm{pb}}\left([\phi_{\mathrm{d}} \quad \theta_{\mathrm{d}} \quad \psi_{\mathrm{d}}]^{\mathrm{T}} - [\phi \quad \theta \quad \psi]^{\mathrm{T}}\right) + \\ \qquad \boldsymbol{k}_{\mathrm{db}}\left(\boldsymbol{\omega}_{\mathrm{B,d}} - \boldsymbol{\omega}_{\mathrm{B}}\right), \end{cases} \tag{15} $$

where ϕ, θ, $\psi \in \mathbb{R}$ are the roll, pitch, and yaw angles of the base, respectively, and $\boldsymbol{k}_{\mathrm{pa}}$, $\boldsymbol{k}_{\mathrm{da}}$, $\boldsymbol{k}_{\mathrm{pb}}$, $\boldsymbol{k}_{\mathrm{db}} \in \mathbb{R}^{3\times3}$ are the proportional and derivative coefficient matrices, respectively.

As more than 70% of the robot mass is concentrated in the base, the robot can be simplified as a single rigid body, in which the CoM coincides with the geometric center of the body (di Carlo et al., 2018). We further assume that the feet cannot generate moments when contacting the ground and that the GRFs are the only external forces acting on the robot.

Based on the force analysis of the single-rigid-body model, the GRFs $\boldsymbol{f}\left(f_x, f_y, f_z\right) \in \mathbb{R}^{3n\times1}$ can be calculated by

$$ \underbrace{\begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \boldsymbol{r}_1\times & \cdots & \boldsymbol{r}_n\times \end{bmatrix}}_{\boldsymbol{A}} \underbrace{[\boldsymbol{f}_1 \quad \cdots \quad \boldsymbol{f}_n]^{\mathrm{T}}}_{\boldsymbol{f}} = \underbrace{\begin{bmatrix} m\left(\boldsymbol{a}_{\mathrm{com,d}} + \boldsymbol{g}\right) \\ \boldsymbol{I}_{\mathrm{G}} \cdot \dot{\boldsymbol{\omega}}_{\mathrm{B,d}} \end{bmatrix}}_{\boldsymbol{b}}, \tag{16} $$

where $m \in \mathbb{R}$ is the total mass of the robot, $\boldsymbol{g} \in \mathbb{R}^3$ denotes the gravity acceleration, $\boldsymbol{I}_{\mathrm{G}} \in \mathbb{R}^{3\times3}$ is the inertia of the CoM, $\boldsymbol{r}_i \in \mathbb{R}^3$ is the vector from the CoM to the foot position of the ith leg (in world coordinates), $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the unit diagonal matrix, and n is the number of stance legs.

3.5.2 GRF distribution

Eq. (16) constitutes six equations with 6–12 (depending on gait) unknown values of $\boldsymbol{f}$. Therefore, Eq. (16) can have an infinite number of solutions. Under certain constraints, the problem of solving this simultaneous set of equations can be transformed into an optimization problem of quadratic programming.

We set the following objective function:

$$ \begin{gathered} \boldsymbol{f}_{\mathrm{d}} = \min_{\boldsymbol{f} \in \mathbb{R}^{12}} \left((\boldsymbol{A}\boldsymbol{f} - \boldsymbol{b})^{\mathrm{T}} \boldsymbol{S} (\boldsymbol{A}\boldsymbol{f} - \boldsymbol{b}) + \delta \boldsymbol{f}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{f} \right), \\ \boldsymbol{d}_{\min} \leqslant \boldsymbol{c}\boldsymbol{f} \leqslant \boldsymbol{d}_{\max}, \end{gathered} \tag{17} $$

where $\boldsymbol{S} \in \mathbb{R}^{6\times6}$ and $\boldsymbol{W} \in \mathbb{R}^{12\times12}$ are positive definite weight matrices, $\delta \in \mathbb{R}$ is the weight of the secondary objective, $\boldsymbol{c} \in \mathbb{R}^{p\times12}$ is the inequality constraint matrix, and $\boldsymbol{d}_{\max}$, $\boldsymbol{d}_{\min} \in \mathbb{R}^p$ are the upper and lower bounds of the constraint, respectively. Here, p denotes the number of inequality constraints.

From an energy-consumption perspective, motion efficiency is usually measured as the transport cost (CoT) (Zhou et al., 2021):

$$ \mathrm{CoT} = \frac{P}{mgv}, \tag{18} $$

where P is the electrical power, and v is the locomotion velocity of the robot.

CoT is proportional to the electric power P when the locomotion velocity is constant. Moreover, under normal working conditions, P is proportional to the joint torque $\boldsymbol{\tau}$, and a corresponding relationship exists between the joint torques and GRFs (see later Eq. (25)). Therefore, penalizing the GRFs means implicitly minimizing the CoT.

Standard quadratic programming is given by

$$\min_{f} \frac{1}{2} \boldsymbol{f}^{\mathrm{T}} \boldsymbol{Hf} + \boldsymbol{f}^{\mathrm{T}} \boldsymbol{g}, \quad \boldsymbol{d}_{\min} \leqslant \boldsymbol{cf} \leqslant \boldsymbol{d}_{\max}, \tag{19}$$

where

$$\begin{cases} \boldsymbol{H} = 2(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{SA} + \delta \boldsymbol{W}), \\ \boldsymbol{g} = -2(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{Sb}), \end{cases} \tag{20}$$

under the following constraints.

(1) Constraints on the upper and lower bounds of the GRFs:

$$f_{\min} \leqslant f_{x,y,z} \leqslant f_{\max}. \tag{21}$$

(2) Constraints by the friction cone (approximated as linear friction to simplify the solution):

$$\begin{cases} -\mu f_z \leqslant |f_x| \leqslant \mu f_z, \\ -\mu f_z \leqslant |f_y| \leqslant \mu f_z, \end{cases} \tag{22}$$

where μ is the friction coefficient.

(3) Zero GRFs during the swing phase:

$$\begin{cases} f_{i(x,y,z)} \neq 0, & S_i = 1, \\ f_{i(x,y,z)} = 0, & S_i = 0. \end{cases} \tag{23}$$

3.5.3 Desired joint torques

The desired joint torques are obtained by summing the feedforward and feedback terms (Boaventura et al., 2012):

$$\boldsymbol{\tau}_{\mathrm{d}} = \boldsymbol{\tau}_{\mathrm{ff}} + \boldsymbol{\tau}_{\mathrm{fb}}. \tag{24}$$

The feedforward torques are generated by mapping the GRFs $\boldsymbol{f}_{\mathrm{d}} \in \mathbb{R}^{12}$ to the joint space:

$$\boldsymbol{\tau}_{\mathrm{ff}} = -\boldsymbol{S}_{\tau} \boldsymbol{J}^{\mathrm{T}} \boldsymbol{f}_{\mathrm{d}}, \tag{25}$$

where $\boldsymbol{S}_{\tau} \in \mathbb{R}^{12\times12}$ is the selection matrix, and $\boldsymbol{J}^{\mathrm{T}} \in \mathbb{R}^{12\times12}$ is the transpose of the Jacobian matrix.

The feedback torques are computed as

$$\boldsymbol{\tau}_{\mathrm{fb}} = \boldsymbol{k}_{\mathrm{p}}(\boldsymbol{q}_{\mathrm{d}} - \boldsymbol{q}) + \boldsymbol{k}_{\mathrm{d}}(\dot{\boldsymbol{q}}_{\mathrm{d}} - \dot{\boldsymbol{q}}), \tag{26}$$

where $\boldsymbol{k}_{\mathrm{p}}$, $\boldsymbol{k}_{\mathrm{d}} \in \mathbb{R}^{12\times12}$ denote the diagonal proportionality matrix and derivative gain matrix, respectively, and $\boldsymbol{q}_{\mathrm{d}}$, $\dot{\boldsymbol{q}}_{\mathrm{d}}$, $\boldsymbol{q}$, $\dot{\boldsymbol{q}} \in \mathbb{R}^{12}$ are the desired and actual joint positions and velocities, respectively.

When the ith leg is in the stance phase (S_i=1), the gains $\boldsymbol{k}_{\mathrm{p}}$ and $\boldsymbol{k}_{\mathrm{d}}$ in Eq. (26) are set low, so the torques $\boldsymbol{\tau}_{\mathrm{d}}$ are mainly generated by the feedforward terms $\boldsymbol{\tau}_{\mathrm{ff}}$. When the ith leg is in the swing phase (S_i=0), $\boldsymbol{\tau}_{\mathrm{ff}}$=0 and the gains in Eq. (26) are set high, so the torques $\boldsymbol{\tau}_{\mathrm{d}}$ are entirely generated by the feedback terms $\boldsymbol{\tau}_{\mathrm{fb}}$.

4 Simulation and experiment

The stability of the locomotion-control strategy, based on dynamic planning of the CoM, was evaluated through simulations and experiments. We performed co-simulations in Matlab (R2018b) and V-REP (version 3.6.2). All experiments were implemented on the prototype shown in Fig. 1c, whose structure imitates the MIT Cheetah 3 robot. The controller parameters are listed in Table 2.

Table 2 Controller parameters

Parameter	Value
Friction coefficient, μ	0.6
CoM proportional gain, $\boldsymbol{k}_{\mathrm{pa}}$ (N/m)	diag(50, 50, 50)
CoM derivative gain, $\boldsymbol{k}_{\mathrm{da}}$ (s·N/m)	diag(10, 10, 10)
Base proportional gain, $\boldsymbol{k}_{\mathrm{pb}}$ (N·m/rad)	diag(50, 50, 50)
Base derivative gain, $\boldsymbol{k}_{\mathrm{db}}$ (s·N·m/rad)	diag(10, 10, 10)
Joint stiffness gain (swing phase), $\boldsymbol{k}_{\mathrm{p}}$ (high) (N·m/rad)	diag(100, 100, 100)
Joint damping gain (swing phase), $\boldsymbol{k}_{\mathrm{d}}$ (high) (s·N·m/rad)	diag(12, 10, 7)
Joint stiffness gain (stance phase), $\boldsymbol{k}_{\mathrm{p}}$ (low) (N·m/rad)	diag(10, 10, 10)
Joint damping gain (stance phase), $\boldsymbol{k}_{\mathrm{d}}$ (low) (s·N·m/rad)	diag(5, 4.5, 3)
CoM inertia, $\boldsymbol{I}_{\mathrm{G}}$ (kg·m^2)	diag(0.31, 1.09, 1.12)
Objective function weight, $\boldsymbol{S}$	diag(5, 5, 10, 10, 10, 10)
Minimum torque weight, $\boldsymbol{W}$	diag(0.5, 0.5, 0.2, 0.5, 0.5, 0.2, 0.5, 0.5, 0.2, 0.5, 0.5, 0.2)
Second objective weight, δ	10
Stability margin, d (m)	0.05
Desired height of robot, H_{d} (m)	0.315

4.1 Omnidirectional locomotion simulation

With $v_{x,\mathrm{d}}$, $v_{y,\mathrm{d}}$, and $\omega_{z,\mathrm{d}}$ set in Eq. (1), the robot moved along the x- and y-axis directions and rotated around the z-axis, thereby achieving omnidirectional locomotion.

The desired velocities were specified as

$$\boldsymbol{v}_{\mathrm{com,\,d}}=\begin{cases}[\,0.2 \quad 0.1 \quad 0 \quad 0 \quad 0 \quad 0.1\,]^{\mathrm{T}}, & \text{gait}=1,\ \text{walk},\\ [\,0.4 \quad 0.2 \quad 0 \quad 0 \quad 0 \quad 0.2\,]^{\mathrm{T}}, & \text{gait}=2,\ \text{trot}.\end{cases} \tag{27}$$

The simulation snapshots of the robot captured during omnidirectional locomotion are shown in Fig. 7. Whether walking (Fig. 7a) or trotting (Fig. 7b), the robot moved stably with the desired CoM velocity.

Fig. 8 shows the GRFs and joint torques of the RF leg during omnidirectional walking (Fig. 8a) and trotting (Fig. 8b). In the stance phase, the GRFs of the RF leg were mainly concentrated in the z direction and were very small in the x and y directions. In contrast, during the swing phase of the RF leg, the GRFs were zero because the foot was above the ground. Additionally, when the RF leg was in the stance phase, the desired torques were mainly dominated by the feedforward terms, and the feedback terms were very small. When the RF leg was in the swing phase, the desired torques were wholly generated by the feedback terms because the feedforward terms were zero. These simulation results were consistent with the design scheme described in Section 3.5.

Fig. 9 shows the joint positions and velocities for the HAA, HFE, and KFE of the RF leg during walking (Figs. 9a and 9b) and trotting (Figs. 9c and 9d). The tracking errors between the desired and simulated values of each joint position and velocity were very small in both gaits, indicating that the simulated results matched the expected results.

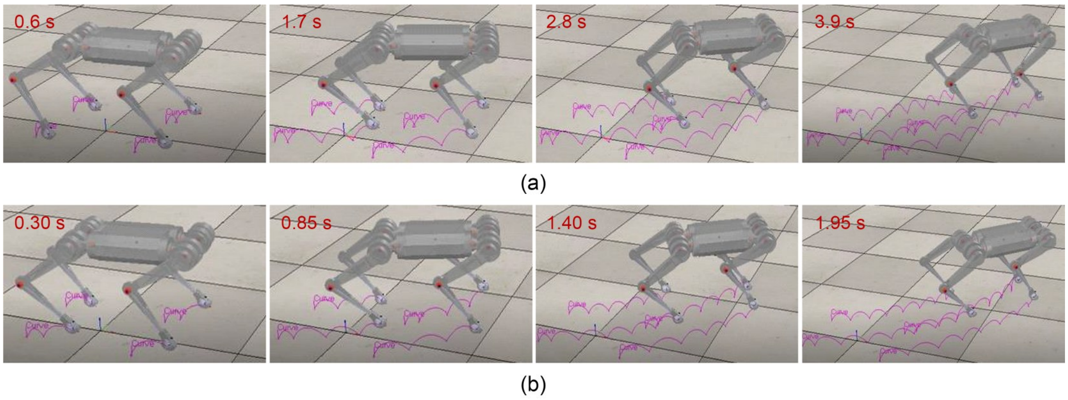
4.2 Omnidirectional locomotion experiment

Fig. 10 shows snapshots of the walking (Fig. 10a) and trotting (Fig. 10b) experiments, with the desired velocities set with Eq. (27). The robot completed the omnidirectional locomotion experiments in both gaits.

Fig. 11 presents the tracking of the CoM position and base orientation during omnidirectional walking and trotting. Figs. 11a and 11c indicate the positions of CoM in the x, y, and z directions in the two gaits, respectively. Figs. 11b and 11d indicate the roll, pitch, and yaw angles of the base in the two gaits, respectively. The simulated CoM and experimental CoM moved to the desired positions, and the errors in base orientation were very small. The positional and orientational tracking errors were slightly larger in the experiments than in the simulations in both gaits. Overall, the results matched expectations, verifying the correctness and effectiveness of the control strategy with CoM dynamic planning.

4.3 Tracking-error analysis

The robot moved at the desired CoM velocity (0.1 or 0.2 m/s along the x-axis in the walk gait; 0.2 or 0.4 m/s along the x-axis in the trot gait). The base orientations were observed during the experiments, and the tracking errors at the different velocities are depicted in Fig. 12. Presented are the major ranges and limits of the variations in the base orientations during walking (Fig. 12a) and trotting (Fig. 12b). Tracking fluctuations for Euler angles and root mean square errors (RMSEs) of the base orientations (Fig. 12c) increased slightly at higher velocities. The control capacity of



(a)

(b)

Fig. 7 Simulation snapshots of omnidirectional locomotion: (a) omnidirectional walking; (b) omnidirectional trotting

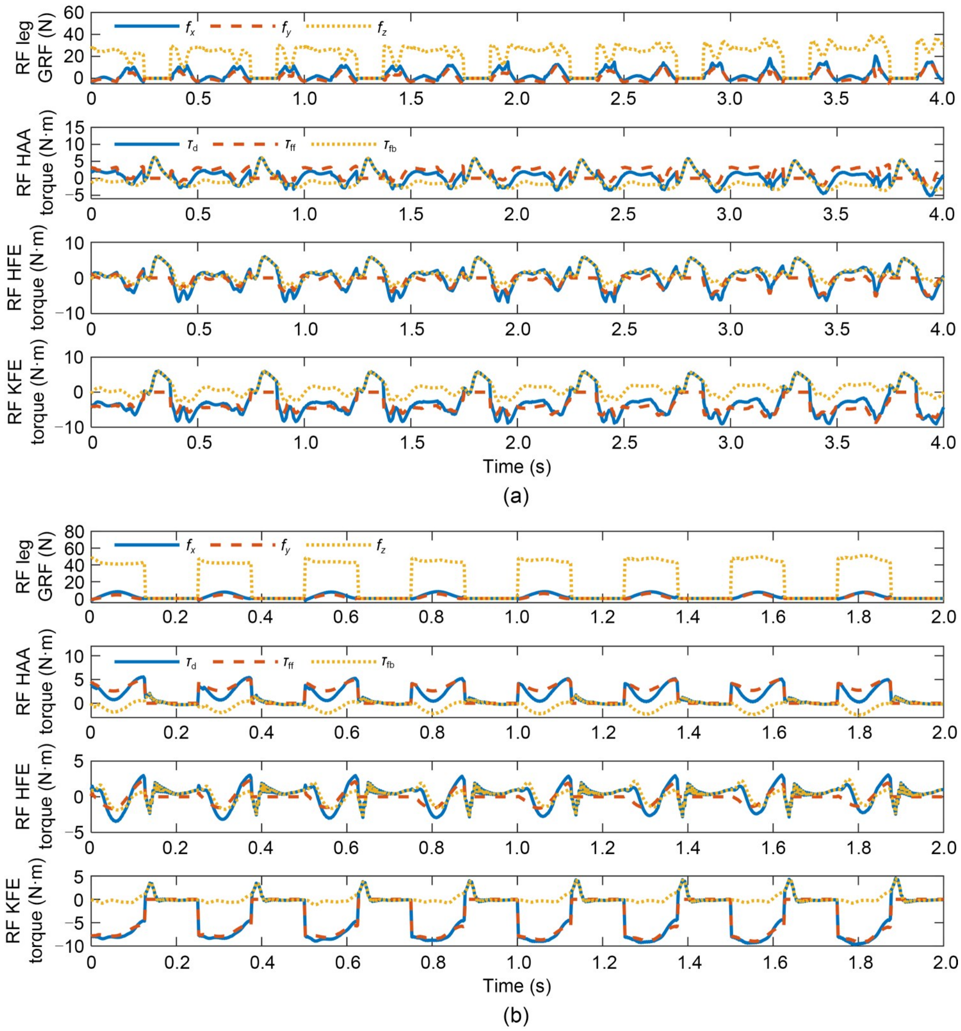


Fig. 8 GRFs and joint torques of RF leg in omnidirectional locomotion: (a) during walking; (b) during trotting

the robot remained stable at different speeds, and the experimental results continued to align with the expected results.

4.4 Comparison of stability and energy consumption

This subsection compares the stability of the robot under the control strategies with and without CoM planning. Here, the robot was governed to walk or trot at $v_{x,d}$=0.2 m/s. Fig. 13 shows the real-time stability margins for the robot during walking (Fig. 13a) and trotting (Fig. 13b).

The body was steady when the stability margin was positive, meaning that the CoM remained in the support triangle. Conversely, the body was unsteady when the stability margin was negative (i.e., the CoM was outside the support triangle). When the support triangle changed during a phase switch in the designed control strategy, the stability margin became negative for a very brief period before recovering (indicating a quick return of the CoM to the support triangle interior). Furthermore, after including CoM planning in the control strategy, the average stability margin and proportion of positive to negative stability margins (Fig. 13c) were improved by 0.0153 m and 27.25%, respectively, during walking, and by 0.0378 m and 37.25%, respectively, during trotting.

Finally, we compared the energy consumption of the robot when the objective function included or excluded ECO during walking and trotting at $v_{x,d}$=0.2 m/s. The real-time instantaneous powers and the mean GRFs (in the z direction) are shown in Fig. 14.

As shown in Fig. 14a, the control strategy with ECO in the objective function yielded a lower mean f_z (GRF in the z direction) than the control strategy without

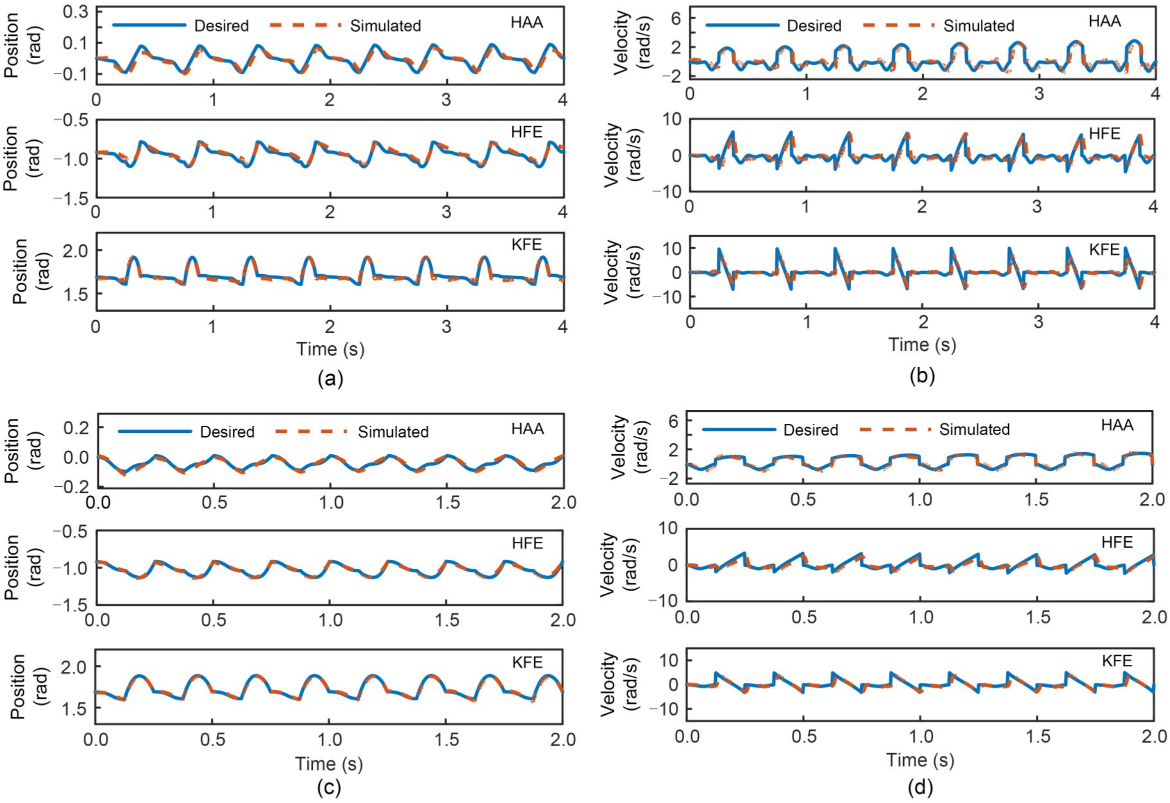


Fig. 9 Leg-joint positions and velocities of RF leg in omnidirectional motion: (a) leg-joint positions during walking; (b) leg-joint velocities during walking; (c) leg-joint positions during trotting; (d) leg-joint velocities during trotting

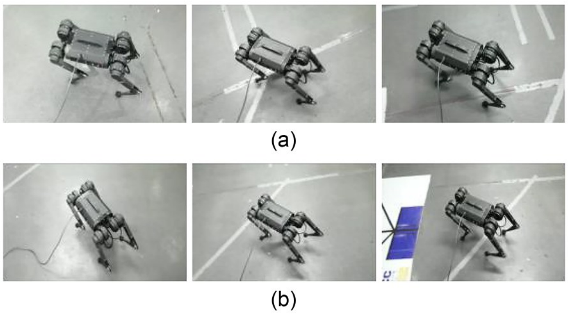


Fig. 10 Experimental snapshots of omnidirectional locomotion: (a) omnidirectional walking; (b) omnidirectional trotting

ECO. The comparisons were more evident in trotting than in walking; specifically, the GRFs decreased by 13.42% and 24.45% during walking and trotting, respectively. As shown in Figs. 14b and 14c, the instantaneous power-variation trends of the robot for the strategies with and without ECO were the same. The ECO reduced the GRFs of the legs in the stance phase, thereby decreasing the corresponding joint torques and the instantaneous power for the robot. Consequently, the control strategy with ECO reduced the average power by 11.25% and 13.83% during walking and trotting, respectively.

5 Conclusions

This paper proposes a novel control strategy with dynamic CoM planning for a quadruped robot. Comparing the control-scheme results with and without CoM planning, we demonstrated that the designed dynamic planning based on synchronous cooperation between the CoM and the swing legs could effectively improve the locomotion stability of the robot. When combined with ECO, the proposed control strategy reduces the locomotion energy consumption of the robot to a certain extent. The control strategy also extends the adaptation range of stability from the walk gait to the trot gait. The effectiveness of the designed control system was verified in simulation and experimental results for omnidirectional locomotion. Tracking-error analysis demonstrated that the controller operates well and enables steady locomotion at different velocities.

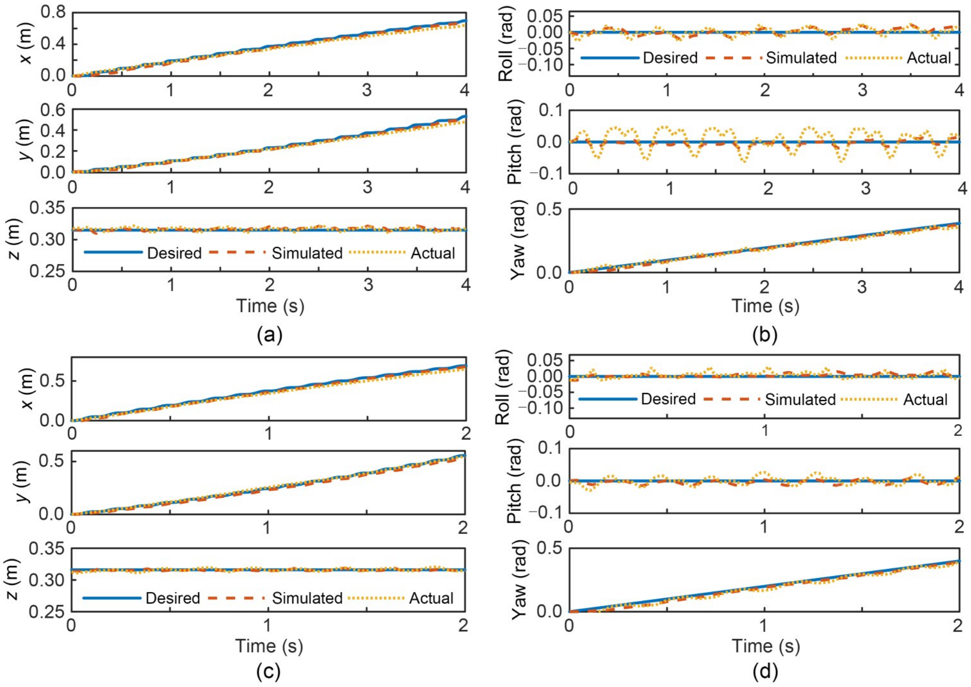


Fig. 11 Tracking of CoM positions and base orientations in omnidirectional locomotion: (a) CoM positions during walking; (b) base orientations during walking; (c) CoM positions during trotting; (d) base orientations during trotting. Solid, dashed, and dotted lines indicate the desired, simulated, and experimental results, respectively

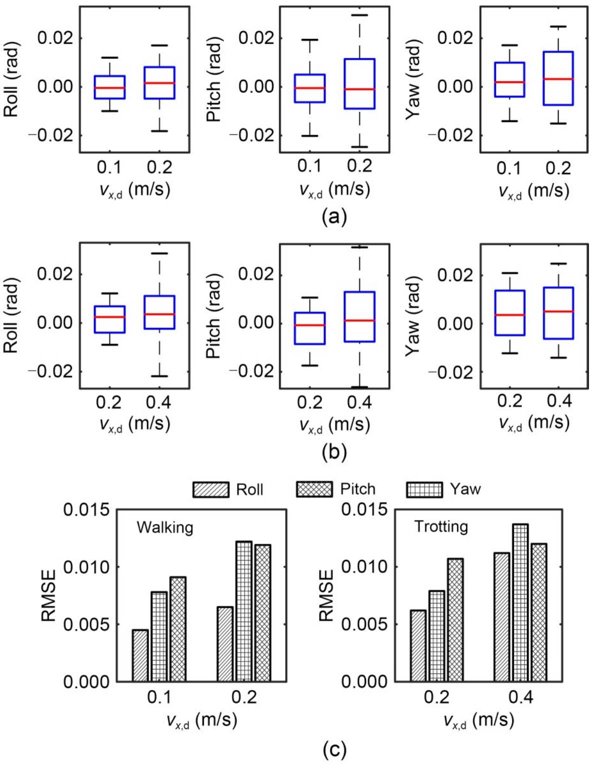


Fig. 12 Tracking errors of the base orientation: (a) boxplots of base orientation during walking; (b) boxplots of base orientation during trotting; (c) bar plots of the tracking errors (RMSE) for both walk and trot gaits

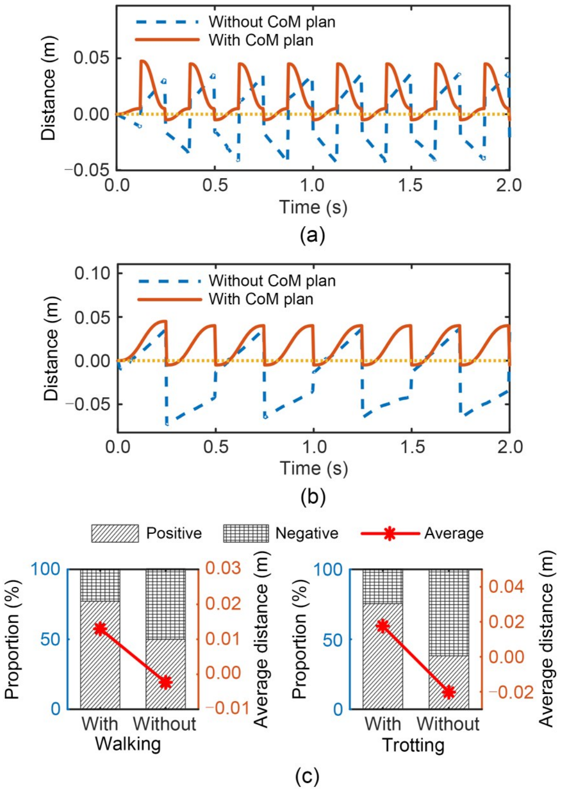


Fig. 13 Stability margins with two control strategies: (a) during walking; (b) during trotting; (c) proportions and averages of stability margins

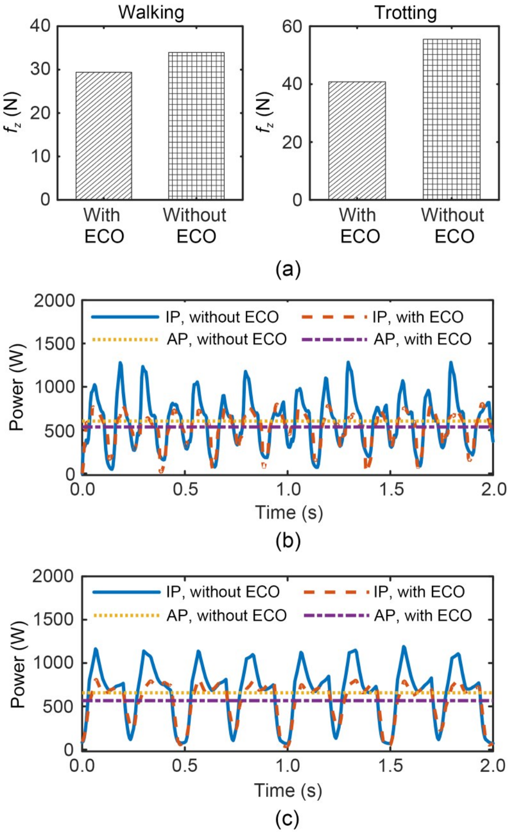


Fig. 14 Comparison between control strategies with and without ECO: (a) bar plots of mean f_z; (b) power during walking; (c) power during trotting. IP: instantaneous power; AP: average power

The control strategy was applied for the typical walk and trot gaits. Because the velocities of the CoM and gait type are pre-programmed, the robot cannot select its gait or velocity in an actual situation. Moreover, the robot was tested in a pre-set experimental scene rather than a natural terrain environment.

In the near future, we plan to incorporate the feedback network of an animal motor nervous system, as well as biological reflexes, into the control system to improve the bionic ability and intelligence of our prototype. We hope that a control system based on multi-level reflex modes will enhance the adaptive control ability of quadruped robots in unstructured terrain.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 52175050 and 52205059), the Outstanding Youth Science Foundation (No. 51922093), the Scientific Research Fund of Zhejiang Provincial Education Department (No. Y202148352), and the Graduate Innovation Special Fund Project of Jiangxi Province (No. YC2021-B031), China.

Author contributions

Yangyang HAN designed the research. Liyu GAO and Huaizhi ZONG processed the corresponding data. Yangyang HAN and Zhenyu LU wrote the first draft of the manuscript. Feifei ZHONG helped to organize the manuscript. Guoping LIU and Junhui ZHANG revised and edited the final version.

Conflict of interest

Yangyang HAN, Guoping LIU, Zhenyu LU, Huaizhi ZONG, Junhui ZHANG, Feifei ZHONG, and Liyu GAO declare that they have no conflict of interest.

References

Arena P, Patanè L, Sueri P, et al., 2021. A data-driven neural network model predictive steering controller for a bio-inspired quadruped robot. *IFAC-PapersOnLine*, 54(17):93-98.
https://doi.org/10.1016/j.ifacol.2021.11.031

Boaventura T, Semini C, Buchli J, et al., 2012. Dynamic torque control of a hydraulic quadruped robot. Proceedings of the IEEE International Conference on Robotics and Automation, p.1889-1894.
https://doi.org/10.1109/ICRA.2012.6224628

Chignoli M, Wensing PM, 2020. Variational-based optimal control of underactuated balancing for dynamic quadrupeds. *IEEE Access*, 8:49785-49797.
https://doi.org/10.1109/ACCESS.2020.2980446

di Carlo J, Wensing PM, Katz B, et al., 2018. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, p.1-9.
https://doi.org/10.1109/IROS.2018.8594448

Ding YR, Pandala A, Li CZ, et al., 2021. Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, 37(4):1154-1171.
https://doi.org/10.1109/TRO.2020.3046415

Dudzik T, Chignoli M, Bledt G, et al., 2020. Robust autonomous navigation of a small-scale quadruped robot in real-world environments. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, p.3664-3671.
https://doi.org/10.1109/IROS45743.2020.9340701

Focchi M, del Prete A, Havoutis I, et al., 2017. High-slope terrain locomotion for torque-controlled quadruped robots. *Autonomous Robots*, 41(1):259-272.
https://doi.org/10.1007/s10514-016-9573-1

Fukui T, Fujisawa H, Otaka K, et al., 2019. Autonomous gait transition and galloping over unperceived obstacles of a quadruped robot with CPG modulated by vestibular feedback. *Robotics and Autonomous Systems*, 111:1-19.
https://doi.org/10.1016/j.robot.2018.10.002

Gonzalez de Santos P, Jimenez MA, Armada MA, 1998. Dynamic effects in statically stable walking machines. *Journal of Intelligent and Robotic Systems*, 23(1):71-85.
https://doi.org/10.1023/A:1007993923530

Gonzalez C, Barasuol V, Frigerio M, et al., 2020. Line walking and balancing for legged robots with point feet. Proceedings of the IEEE/RSJ International Conference on

Intelligent Robots and Systems, p.3649-3656.
https://doi.org/10.1109/IROS45743.2020.9341743

Hao Q, Wang ZB, Wang JZ, et al., 2020. Stability-guaranteed and high terrain adaptability static gait for quadruped robots. *Sensors*, 20(17):4911.
https://doi.org/10.3390/s20174911

Hutter M, Gehring C, Jud D, et al., 2016. ANYmal–a highly mobile and dynamic quadrupedal robot. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, p.38-44.
https://doi.org/10.1109/IROS.2016.7758092

Hutter M, Gehring C, Lauber A, et al., 2017. ANYmal–toward legged robots for harsh environments. *Advanced Robotics*, 31(17):918-931.
https://doi.org/10.1080/01691864.2017.1378591

Hyun DJ, Seok S, Lee J, et al., 2014. High speed trot-running: implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah. *The International Journal of Robotics Research*, 33(11):1417-1445.
https://doi.org/10.1177/0278364914532150

Lee C, An D, 2021. Reinforcement learning and neural network-based artificial intelligence control algorithm for self-balancing quadruped robot. *Journal of Mechanical Science and Technology*, 35(1):307-322.
https://doi.org/10.1007/s12206-020-1230-0

Lin PC, Komsuoglu H, Koditschek DE, 2005. A leg configuration measurement system for full-body pose estimates in a hexapod robot. *IEEE Transactions on Robotics*, 21(3): 411-422.
https://doi.org/10.1109/TRO.2004.840898

Liu LQ, Zhang CR, 2020. Dynamic properties of VDP-CPG model in rhythmic movement with delay. *Mathematical Biosciences and Engineering*, 17(4):3190-3202.
https://doi.org/10.3934/mbe.2020181

McClain EW, Meek S, 2018. Determining optimal gait parameters for a statically stable walking human assistive quadruped robot. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, p.1751-1756.
https://doi.org/10.1109/IROS.2018.8593979

Park HW, Wensing PM, Kim S, 2017. High-speed bounding with the MIT Cheetah 2: control design and experiments. *The International Journal of Robotics Research*, 36(2): 167-192.
https://doi.org/10.1177/0278364917694244

Pepe G, Laurenza M, Belfiore NP, et al., 2021. Quadrupedal robots’ gaits identification via contact forces optimization. *Applied Sciences*, 11(5):2102.
https://doi.org/10.3390/app11052102

Raibert MH, 1986. Legged Robots That Balance. MIT Press, Cambridge, USA, p.44-56.

Shao YC, Jin YB, Liu XW, et al., 2022. Learning free gait transition for quadruped robots via phase-guided controller. *IEEE Robotics and Automation Letters*, 7(2):1230-1237.
https://doi.org/10.1109/LRA.2021.3136645

Srinivas T, Madhusudhan AKK, Manohar L, et al., 2021. Valkyrie-design and development of gaits for quadruped robot using particle swarm optimization. *Applied Sciences*, 11(16):7458.
https://doi.org/10.3390/app11167458

Tian J, Ma C, Wei C, et al., 2019. A smooth gait planning framework for quadruped robot based on virtual model control. Proceedings of the 12th International Conference on Intelligent Robotics and Applications, p.398-410.
https://doi.org/10.1007/978-3-030-27538-9_34

Wang YQ, Ye LQ, Wang XQ, et al., 2020. A static gait generation for quadruped robots with optimized walking speed. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, p.1899-1906.
https://doi.org/10.1109/SMC42975.2020.9282997

Yeom H, Bae J, 2021. A dynamic gait stabilization algorithm for quadrupedal locomotion through contact time modulation. *Nonlinear Dynamics*, 104(3):2275-2289.
https://doi.org/10.1007/s11071-021-06376-5

Zhang ML, Zhang YJ, He XL, et al., 2021. Adaptive pid control and its application based on a double-layer BP neural network. *Processes*, 9(8):1475.
https://doi.org/10.3390/pr9081475

Zhang SS, Liu M, Yin YF, et al., 2019. Static gait planning method for quadruped robot walking on unknown rough terrain. *IEEE Access*, 7:177651-177660.
https://doi.org/10.1109/ACCESS.2019.2958320

Zhang Y, Wang H, Ding Y, et al., 2021. Adaptive walking control for a quadruped robot on irregular terrain using the complex-valued CPG network. *Symmetry*, 13(11):2090.
https://doi.org/10.3390/sym13112090

Zhou LL, Li TF, Liu ZY, et al., 2021. An efficient gait-generating method for electrical quadruped robot based on humanoid power planning approach. *Journal of Bionic Engineering*, 18(6):1463-1474.
https://doi.org/10.1007/s42235-021-00089-6