J. Shanghai Jiaotong Univ. (Sci.), 2014, 19(1): 1-10
DOI: 10.1007/s12204-014-1470-1

Central Pattern Generator Based Gait Control for Planar Quadruped Robots

LI Jia-wang (李家旺), *WU Chao* (吴 超), *GE Tong** (葛 彤)
(School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: In this paper, a gait control scheme is presented for planar quadruped robots based on a biologic concept, namely central pattern generator (CPG). A CPG is modeled as a group of the coupled nonlinear oscillators with an interaction weighting matrix which determines the gait patterns. The CPG model, mapping functions and a proportional-differential (PD) joint controller compose the basic gait generator. By using the duty factor of gait patterns as a tonic signal, the activity of the CPG model can be modulated, and as a result, a smooth transition between different gait patterns is achieved. Moreover, by tuning the parameters of the CPG model and mapping functions, the proposed basic gait generator can realize adaptive workspace trajectories for the robot to suit different terrains. Simulation results illustrate and validate the effectiveness of the proposed gait controllers.
Key words: quadruped robot, central pattern generator (CPG), gait transition, adaptation
CLC number: TP 242 **Document code:** A

0 Introduction

In the past several years, the locomotion control problem for legged robots inspired by central pattern generators (CPGs) has received increasing interests and has become an important topic in the robotics community. CPGs are neuron circuits consisting of group neurons that can realize rhythmic activity patterns via some simple inputs which not need to be rhythmic[1]. In general, CPGs are modeled by coupled nonlinear oscillators in various types such as Matsuoka's oscillator[2-3], Hopf oscillator[4-6] and others[7-8]. Moreover, the parameters in these CPG models can be modulated to switch gait patterns or to cope with irregular terrains via the dynamic interaction among the higher-level stimulation system, the oscillators and the sensory reflexes. Some properties of CPG models also make them suitable and reliable for the control of other types of robots besides legged robots, such as swimming robots[9-11] and modular reconfigurable robots[12].

In this paper, we address an issue about the generation of different gait patterns and the smooth transitions between them for planar quadruped robots, namely the planar projections of robots. To this end, we use a modified Hopf oscillator as a CPG model to generate some basic rhythmic motions. The CPG outputs are mapped online to the desired workspace trajectories of the robot legs rather than to the torque control signals[13-14] or phase regulation signals[15] in the joint space, which, as introduced in Ref. [3], makes the generated gaits more realistic and easier to be modulated. Moreover, in this work we extend the idea presented in Ref. [5] for the gait transitions. Herein, we design a simple stimulation system to regulate the activity of CPGs, and as a result, to realize a smooth switching among different gaits. Different from Ref. [5], we use one of the gait parameters instead of an additional variable as a tonic signal of the CPG model, which makes our model more concise.

Another topic addressed in this paper is to realize gait adaptation on irregular terrains for the robot. Many researches have been done so far on this problem. For instance, Matsubara et al.[16] proposed a sensory feedback learning framework for CPG-based biped locomotion. In Ref. [17], the authors performed an adaptive behavior transition between bipedal and quadruped for walking robots, where the robot's posture was regulated based on the neuromodulation mechanism proposed by Ref. [13]. In both works, the CPG outputs were mapped to the target joint angles. As a different choice, a discrete movement generator was presented in Ref. [14] for biped robots to achieve gait adaptation, whereas the output of CPG was used to generate the

Received date: 2011-08-09
Foundation item: the National High Technology Research and Development (863) Program of China (No. 2007AA09Z215), the National Natural Science Foundation of China (No. 51009091) and the Research Fund for the Doctoral Program of Higher Education of China (No. 20100073120016)
∗**E-mail:** tongge@sjtu.edu.cn

muscle torque. For quadruped robots, the authors of Ref. [3] and Ref. [15] realized adaptive walking on irregular terrains for the robot "Tekken" by integrating some sensory reflexes in the CPG model, where the CPG outputs were mapped to the joint angles. In contrast to these works, we perform stable gait adaptation for a quadruped robot by using the modulation of both the CPG model and the robot's kinematics, while the information about the irregular terrain is assumed to be known. Since the CPG outputs are mapped to the workspace trajectories, the effectiveness of the proposed gait adaptation scheme can be easily and straightforwardly validated.

Firstly, this paper discusses issues about the inverse kinematics and dynamics for a planar quadruped robot, and a simple proportional-differential (PD) joint controller scheme is also presented. We introduce a CPG model and mapping functions which are applied as a basic desired workspace trajectory generator during this study. Also, we present the details of modulation of the parameters both in the CPG model and mapping functions for gait transition and adaptation. Some illustrative results are presented. Finally, we summarize this paper and briefly address our future work.

1 Robot Kinetics

1.1 Single Leg Inverse Kinematics

A simplified model of a planar quadruped robot is depicted in Fig. 1. The robot is composed of nine links: a main body and four legs with knee joints. Thus, the robot has 11 degrees of freedom (DoFs). To simplify the analysis, we assume that the thickness of all links is negligible and each link's mass is concentrated at its geometric center. Coordinate Oxy represents an earth-fixed inertial frame. Coordinate $O_3x_3y_3$ represents a local frame fixed to the third leg of the robot, where the origin is located at the corresponding hip joint. Variable ψ_3 denotes the rotation angle of the $O_3x_3y_3$ frame with respect to the Oxy frame. We can also define some local fixed frames attached to the other legs in a similar way: θ_{B} is the pitch angle of the main body; $\boldsymbol{q}_{\mathrm{h}} = [\theta_{\mathrm{h1}}\ \theta_{\mathrm{h2}}\ \theta_{\mathrm{h3}}\ \theta_{\mathrm{h4}}]^{\mathrm{T}}$ and $\boldsymbol{q}_{\mathrm{k}} = [\theta_{\mathrm{k1}}\ \theta_{\mathrm{k2}}\ \theta_{\mathrm{k3}}\ \theta_{\mathrm{k4}}]^{\mathrm{T}}$ denote the rotating angles of the hip and knee joints, respectively; L_{B}, L_{h} and L_{k} are the lengths of the body, upper and lower links, respectively. The dotted curve illustrates the desired workspace trajectory of the one of the fore leg-ends, where S and H are respectively the gait length and the maximum height.

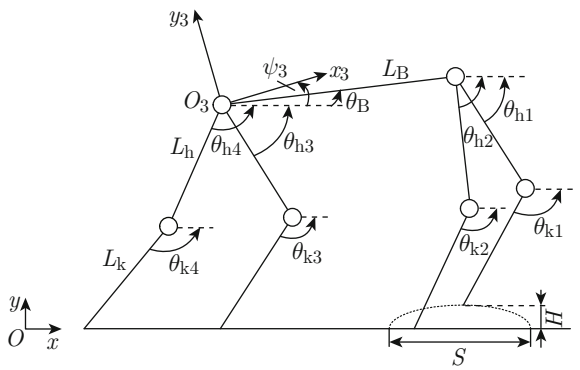


Fig. 1 Schematic of a quadruped robot and parameters

Let $\boldsymbol{\eta}_{\mathrm{h}i} = [x_{\mathrm{h}i}\ y_{\mathrm{h}i}]^{\mathrm{T}}$ and $\boldsymbol{\eta}_{\mathrm{f}i} = [x_{\mathrm{f}i}\ y_{\mathrm{f}i}]^{\mathrm{T}}$ be the position vectors of the ith hip joint and corresponding leg-end with respect to the Oxy frame, respectively. Then we have

$$\boldsymbol{\eta}_{\mathrm{h}i} = \boldsymbol{\eta}_{\mathrm{f}i} + \begin{bmatrix} -\cos\theta_{\mathrm{h}i} & -\cos\theta_{\mathrm{k}i} \\ \sin\theta_{\mathrm{h}i} & \sin\theta_{\mathrm{k}i} \end{bmatrix} \begin{bmatrix} L_{\mathrm{h}} \\ L_{\mathrm{k}} \end{bmatrix}, \quad 1 \leqslant i \leqslant 4. \tag{1}$$

Due to the parameter definition and the fact that the rotating angles of the knee joints are usually smaller than that of the hip joints, i.e. $\theta_{\mathrm{k}i} < \theta_{\mathrm{h}i}$, we can express the inverse kinematics of the ith leg as

$$\left.\begin{aligned} \theta_{\mathrm{h}i} &= -\frac{\pi}{2} - \arctan\left(\frac{x_{\mathrm{h}i} - x_{\mathrm{f}i}}{y_{\mathrm{h}i} - y_{\mathrm{f}i}}\right) + \\ &\qquad \arcsin\left(\frac{L_{\mathrm{k}}\sin\theta_{\mathrm{a}}}{\sqrt{l}}\right) \\ \theta_{\mathrm{k}i} &= \theta_{\mathrm{h}i} - \theta_{\mathrm{a}} \end{aligned}\right\}, \tag{2}$$

where

$$\theta_{\mathrm{a}} = \pi - \arccos\left(\frac{L_{\mathrm{h}}^2 + L_{\mathrm{k}}^2 - l}{2L_{\mathrm{h}}L_{\mathrm{k}}}\right),$$
$$l = (x_{\mathrm{h}i} - x_{\mathrm{f}i})^2 + (y_{\mathrm{h}i} - y_{\mathrm{f}i})^2.$$

1.2 Dynamic Formulation

As shown in Fig. 1, the configuration of the robot can be defined with 11 coordinates,

$$\boldsymbol{q} = [x_{\mathrm{C}}\ y_{\mathrm{C}}\ \theta_{\mathrm{B}}\ \boldsymbol{q}_{\mathrm{h}}^{\mathrm{T}}\ \boldsymbol{q}_{\mathrm{k}}^{\mathrm{T}}] \in \mathbb{R}^{11},$$

where the first two coordinates represent the position of the main body's center of geometry (CG) with respect to the Oxy frame. By using Lagrangian formulation, the resultant dynamic model of the robot can be written in the form:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{\lambda} + \boldsymbol{B}\boldsymbol{\tau}, \tag{3}$$

where, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{11\times 11}$ is the symmetric and positive definite inertia matrix; $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{11}$ and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{11}$ denote the Coriolis/centrifugal and gravity terms, respectively; $\boldsymbol{B} \in \mathbb{R}^{11\times 9}$ is a full-rank constant matrix; $\boldsymbol{\tau} \in \mathbb{R}^9$ is the vector of the actuating torques; $\boldsymbol{\Phi}(\boldsymbol{q}) \in \mathbb{R}^{8\times 11}$ denotes the Jacobian matrix of all leg-ends' position coordinates with respect to the Oxy frame, i.e.,

$$\boldsymbol{\Phi}(\boldsymbol{q}) = \frac{\partial}{\partial \boldsymbol{q}}[x_{\mathrm{f1}}\ y_{\mathrm{f1}}\ \cdots\ x_{\mathrm{f4}}\ y_{\mathrm{f4}}]^{\mathrm{T}}; \tag{4}$$

$\boldsymbol{\lambda} \in \mathbb{R}^8$ consists of the reaction forces acting on the leg-ends from the ground,

$$\boldsymbol{\lambda} = [F_{x1}\ F_{y1}\ \cdots\ F_{x4}\ F_{y4}]^{\mathrm{T}}, \tag{5}$$

F_{xi} and F_{yi} denote respectively the x- and y-directional components of the reaction forces that the ground exerts on the ith leg-end, which are modeled by a virtual spring-damper system,

$$\left.\begin{aligned} F_{xi} &= -K_x(x_{\mathrm{f}i} - x_{0i}) - D_x\dot{x}_{\mathrm{f}i} \\ F_{yi} &= -K_y(y_{\mathrm{f}i} - y_{0i}) - D_y\dot{y}_{\mathrm{f}i} \end{aligned}\right\}, \quad y_{\mathrm{f}i} < y_{0i}, \tag{6}$$

the positive scalars K_i and $D_i\,(i = x, y)$ are the spring and damping constants respectively, y_{0i} denotes the local height of the ground, and x_{0i} is the horizontal position of the ith leg-end as $y_{\mathrm{f}i} = y_{0i}$. Consequently, the components of $\boldsymbol{\lambda}$ are null when the corresponding leg-ends are not contacting with the ground.

1.3 Controller Design

From Fig. 1, we know that, if the desired trajectories of the position vectors $\boldsymbol{\eta}_{\mathrm{h}i}$ and $\boldsymbol{\eta}_{\mathrm{f}i}$ are determined, all of the joint angles $\theta_{\mathrm{h}i}$ and $\theta_{\mathrm{k}i}$ $(1 \leqslant i \leqslant 4)$ can be calculated by the inverse kinematics Eq. (2), and consequently, the robot's configuration is specified. Therefore, the resultant control structure for the robot consists of an input controller and a gait generator, where the latter one is to generate the desired trajectories of the hip and leg-end positions, namely $\boldsymbol{\eta}^{\mathrm{d}}_{\mathrm{h}i}$ and $\boldsymbol{\eta}^{\mathrm{d}}_{\mathrm{f}i}$. The control diagram is illustrated in Fig. 2, in which $\boldsymbol{q}^{\mathrm{d}}$ is the vector of desired generalized coordinates.

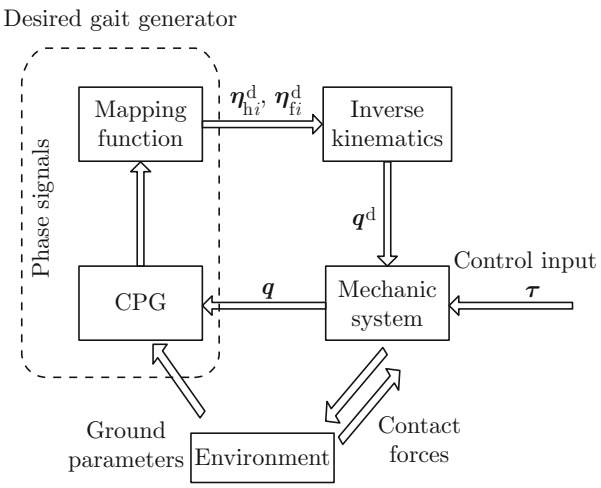


Fig. 2 Control diagram

In order to design a control law for the actuating torques $\boldsymbol{\tau}$, one should note that the robot is an underactuated system since it has 11 DoFs and has only 9 inputs. To simplify the controller design procedure, we use the reduced dynamics method. For this purpose, we first divide the vector of generalized coordinates $\boldsymbol{q}$ into two parts, namely $\boldsymbol{q} = [\boldsymbol{q}_{\mathrm{P}}^{\mathrm{T}}\ \boldsymbol{q}_{\mathrm{A}}^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{q}_{\mathrm{A}} = [\theta_{\mathrm{B}}\ \boldsymbol{q}_{\mathrm{h}}^{\mathrm{T}}\ \boldsymbol{q}_{\mathrm{k}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^9$ and $\boldsymbol{q}_{\mathrm{P}} = [x_{\mathrm{C}}\ y_{\mathrm{C}}]^{\mathrm{T}} \in \mathbb{R}^2$ denote the vectors of active and passive states, respectively. Generally, the passive states $\boldsymbol{q}_{\mathrm{P}}$ can be specified by the active states $\boldsymbol{q}_{\mathrm{A}}$ and the inverse kinematics. In this paper, we will propose a control law for the active states $\boldsymbol{q}_{\mathrm{A}}$ only. Denote $\boldsymbol{q}_{\mathrm{A}} = \boldsymbol{H}\boldsymbol{q}$ where $\boldsymbol{H} \in \mathbb{R}^{9\times 11}$ is a time-invariant mapping matrix. The reduced dynamics from Eq. (3) can be written as

$$\ddot{\boldsymbol{q}}_{\mathrm{A}} = \boldsymbol{H}\boldsymbol{M}^{-1}(\boldsymbol{q})[-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{\lambda} + \boldsymbol{B}\boldsymbol{\tau}]. \tag{7}$$

Then, by using a simple PD control law, the actuating torques are given by

$$\boldsymbol{\tau} = (\boldsymbol{H}\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{B})^{-1}[K_\theta(\boldsymbol{q}_{\mathrm{A}}^{\mathrm{d}} - \boldsymbol{q}_{\mathrm{A}}) - D_\theta\dot{\boldsymbol{q}}_{\mathrm{A}}], \tag{8}$$

where $\boldsymbol{q}_{\mathrm{A}}^{\mathrm{d}}$ represents the desired active states, and the positive scales K_θ and D_θ are respectively the proportional and derivative gains to be chosen later. One can see that, if the control gains K_θ and D_θ are sufficiently large, the active state vector $\boldsymbol{q}_{\mathrm{A}}$ will track the desired one as closely as possible.

2 Generation of Desired Gait Patterns via CPG

In this section, we will present a desired gait generator for the robot by using CPG concepts. As shown in Fig. 2, the gait generator consists of a CPG and a mapping function. The CPG model is used to generate some interconnected and stable oscillated phase signals. Then, the mapping function transforms these signals into the robot's desired workspace trajectories. Moreover, since the interconnection of CPG signals can be changed via different CPG network structures, this property can be used to realize various gait patterns for the robot.

2.1 CPG Model and Network Structures

The rhythmic movement of the robot can be seen as proper and stable movement sequences of four legs. To generate rhythmic signal for each leg, we apply a modified form of the dynamic oscillators introduced in Ref. [4] as a unit CPG model, which is given by

$$\left.\begin{aligned} \dot{u}_i &= A_u(\mu - r_i^2)u_i + \omega_i v_i \\ \dot{v}_i &= A_v(\mu - r_i^2)v_i - \omega_i u_i + \sum_{j\neq i}^{n} k_{ij}v_j \\ \zeta_i &= u_i + S_i \end{aligned}\right\}, \tag{9}$$

where, $r_i = \sqrt{u_i^2 + v_i^2}$, the subscript i denotes the index of oscillators; u_i and v_i are the inner states of the ith oscillator; ζ_i and ω_i are the output and frequency of the ith oscillator, respectively; μ is an excitation signal; A_u and A_v are positive constants controlling the convergence speed of the inner states; k_{ij} represents the

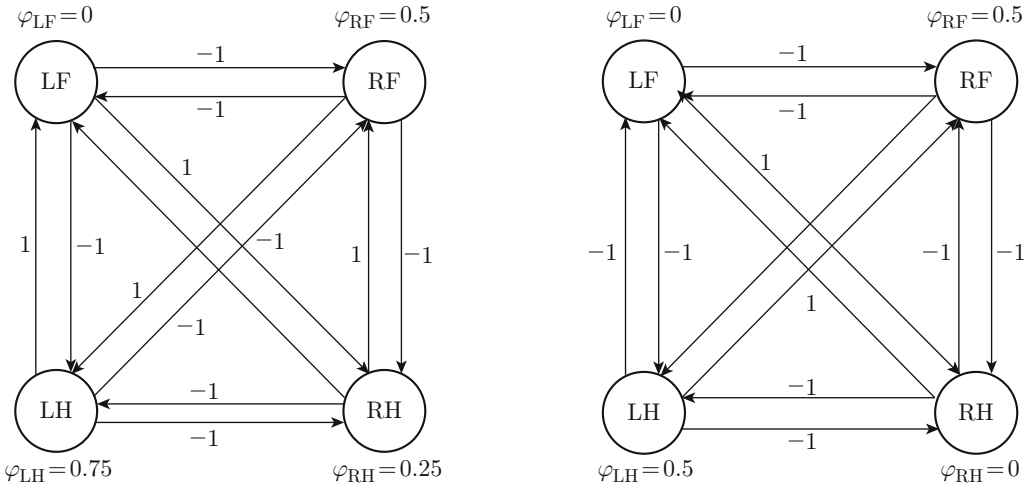
weight factor of connection from the jth oscillator to the ith oscillator; S_i is the feedback signal from the external environments to the ith oscillator. Here we add the feedback term S_i directly on the oscillator output ζ_i rather than on the inner states since it can assure more expectable and realistic reactions of the oscillator which affected by external disturbances.

Note that the oscillator described by Eq. (9) generates periodic trajectories of the inner states where the ascending and descending parts have the same durations. However, for each leg movement of a quadruped robot, the durations of the swing and stance phases maybe different form each other. Due to this fact, we use the following equation proposed in Ref. [4] which can make the frequency to switch smoothly between two different values:

$$\omega_i = \frac{\omega_{\rm st}}{1+\mathrm{e}^{bv_i}} + \frac{\omega_{\rm sw}}{1+\mathrm{e}^{-bv_i}}, \tag{10}$$

where $\omega_{\rm st}$ and $\omega_{\rm sw}$ are the frequencies of the stance and swing phases, respectively, and b is constant to be chosen later.

As also mentioned above, every leg is represented by only one oscillator. Therefore, the whole CPG network of a quadruped robot consists of four coupled oscillators, i.e., n= 4 in Eq. (9). The mutual effects of these oscillators can be divided into two modes: excitation and inhibition. In general, it is said that the connection from the jth oscillator to the ith oscillator is inhibition if the weight factor $k_{ij} < 0$, vice versa. In Fig. 3, two CPG network structures are plotted for the robot in walk and trot gaits, respectively.



(a) Walk: each leg lagging one quarter of the gait cycle (b) Tort: diagonal legs moving in same phase

Fig. 3 Sketch of walk and tort gait event consequences of quadruped robots

In the figure, indices of the legs are left fore (LF), right fore (RF), left hind (LH) and right hind (RH); φ_i, $i \in$\{LF, RF, LH, RH\}, is the unified relative phase of the ith leg. If the legs are numbered as "LF" := 1, "RF" := 2, "LH" := 3 and "RH" := 4, we can obtain the weighting matrices in walk and tort gaits respectively as

$$\left.\begin{aligned} \boldsymbol{K}_{\rm walk} &= \begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ -1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{bmatrix} \\ \boldsymbol{K}_{\rm tort} &= \begin{bmatrix} 0 & -1 & -1 & 1 \\ -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{bmatrix} \end{aligned}\right\}. \tag{11}$$

However, it should be noted that, when the legs are numbered in different order, the expressions of the matrices in Eq. (11) need to be changed accordingly.

2.2 Mapping Function

In this section, the phase signals of the unit CPG model Eq. (9) are mapped to the desired workspace trajectories of the robot movements. Note that in a complete gait circle, every leg's movement can be divided into two parts: swing phase and stance phase. Therefore, we can also design the mapping functions for these two phases separately.

In the swing phase of the ith leg movement, the legend lifts up in the air from the posterior extreme position (PEP) and falls down to contact the ground at the anterior extreme position (AEP). Note that from Eq. (10) $v_i \geqslant 0$ during the swing phase, and thus, the desired workspace trajectory with respect to the local coordinate frame can be generated by the following

mapping function

$$\boldsymbol{r}_{i,\mathrm{sw}}^{\mathrm{d}} = \boldsymbol{J}(\psi_i)\begin{bmatrix} x_{\mathrm{f}i}^{\mathrm{d}} - x_{\mathrm{h}i} \\ y_{\mathrm{f}i}^{\mathrm{d}} - y_{\mathrm{h}i} \end{bmatrix} = \begin{bmatrix} k_x\zeta_i \\ -y_{\mathrm{h}i,\mathrm{B}} + k_y v_i \end{bmatrix}, \quad \forall v_i \geqslant 0, \tag{12}$$

where, $y_{\mathrm{h}i,\mathrm{B}}$ is the projection of the hip joint's coordinate $y_{\mathrm{h}i}$ with respect to the local frame; k_x and k_y are the amplitude gain coefficients; $\boldsymbol{J}(\psi_i)$ is the coordinate transformation matrix from the Oxy frame to the local frame, which is defined as

$$\boldsymbol{J}(\psi_i) = \begin{bmatrix} \cos\psi_i & \sin\psi_i \\ -\sin\psi_i & \cos\psi_i \end{bmatrix}.$$

Due to the fact that the hip joint's position ($x_{\mathrm{h}i}$, $y_{\mathrm{h}i}$) and the angle ψ_i are all known, then we can calculate the leg-end's desired trajectory $\boldsymbol{\eta}_{\mathrm{f}i}^{\mathrm{d}}$ from Eq. (12). Alternatively, we can also directly calculate the desired angles $\theta_{\mathrm{h}i}^{\mathrm{d}}$ and $\theta_{\mathrm{k}i}^{\mathrm{d}}$ based on Eqs. (2) and (12), which can make the computation more efficient.

In the stance phase, the position of the leg-end is fixed, while the hip joint keeps moving. Therefore, our real objective in this stage is to generate the desired trajectory of the hip joint. Then, following a similar way introduced above, the mapping function for the hip joint in the stance phase can be chosen as

$$\boldsymbol{r}_{i,\mathrm{st}}^{\mathrm{d}} = \boldsymbol{J}(\psi_i)\begin{bmatrix} x_{\mathrm{f}i} - x_{\mathrm{h}i}^{\mathrm{d}} \\ y_{\mathrm{f}i} - y_{\mathrm{h}i}^{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} k_x\zeta_i \\ -y_{\mathrm{h}i,\mathrm{B}}^{\mathrm{d}} \end{bmatrix}, \quad \forall v_i < 0. \tag{13}$$

However, it should be noted that, during either a tort or a walk gait, there always exist at least a fore leg and a hind leg contacting the ground simultaneously, which implies that the movements of the supporting legs are interrelated, and thus, the desired trajectories of these legs given by Eq. (13) maybe conflict with each other. To remove this problem, we introduce a predictor-corrector method. At first, by using the geometric relations plotted in Fig. 1, we choose the following mapping function for the supporting legs:

$$\left.\begin{aligned} x_{\mathrm{C}}^{\mathrm{d}} &= (x_{\mathrm{F,p}}^{\mathrm{d}} + x_{\mathrm{H,p}}^{\mathrm{d}})/2 \\ y_{\mathrm{C}}^{\mathrm{d}} &= (y_{\mathrm{F,p}}^{\mathrm{d}} + y_{\mathrm{H,p}}^{\mathrm{d}})/2 \\ \theta_{\mathrm{B}}^{\mathrm{d}} &= \arctan\left(\frac{y_{\mathrm{F,p}}^{\mathrm{d}} - y_{\mathrm{H,p}}^{\mathrm{d}}}{x_{\mathrm{F,p}}^{\mathrm{d}} - x_{\mathrm{H,p}}^{\mathrm{d}}}\right) \end{aligned}\right\}, \tag{14}$$

where, ($x_{\mathrm{C}}^{\mathrm{d}}$, $y_{\mathrm{C}}^{\mathrm{d}}$) is the desired position of CG; $\theta_{\mathrm{B}}^{\mathrm{d}}$ is the desired pitch angle of the main body; ($x_{\mathrm{F,p}}^{\mathrm{d}}$, $y_{\mathrm{F,p}}^{\mathrm{d}}$) and ($x_{\mathrm{H,p}}^{\mathrm{d}}$, $y_{\mathrm{H,p}}^{\mathrm{d}}$) denote respectively the predicted trajectories of the fore and hind hip joints given by Eq. (13). Then, we can obtain the corrected trajectory of the ith hip joint in the stance phase by

$$\left.\begin{aligned} x_{\mathrm{h}i}^{\mathrm{d}} &= x_{\mathrm{C}}^{\mathrm{d}} \pm (L_{\mathrm{B}}/2)\cos\theta_{\mathrm{B}}^{\mathrm{d}} \\ y_{\mathrm{h}i}^{\mathrm{d}} &= y_{\mathrm{C}}^{\mathrm{d}} \pm (L_{\mathrm{B}}/2)\sin\theta_{\mathrm{B}}^{\mathrm{d}} \end{aligned}\right\}, \tag{15}$$

where "+" for the fore legs and "−" for the hind legs.

3 Gait Transition and Adaptation

3.1 Gait Patterns Transition

For quadruped robots, one of the most important features is the ability of smooth switching between different gait patterns. To achieve this, and noting that different gait patterns correspond to different sets of CPG parameters, we will use a stimulation system besides the CPG model to initiate and regulate the activity of the CPG networks, as the brain-like control system introduced in Ref. [5]. In our model, a variable β, namely the duty factor, is used as the stimulation signal to adjust the needed parameters of the CPG models. These parameters include the resulting frequency ω_i, the weight factor k_{ij} and the excitation signal μ.

The duty factor β is the radio between the stance phase duration and the cycle time,

$$\beta = T_{\mathrm{st}}/(T_{\mathrm{sw}} + T_{\mathrm{st}}), \tag{16}$$

with T_{st} the duration of the stance phase and T_{sw} the duration of the swing phase. Generally, the value of β is set to $\beta \in [0.5, 1)$. For legged robots, gait frequency ω_i can always be specified by the duty factor β and the swing frequency ω_{sw}. Due to the definitions of ω_{st} and ω_{sw}, we know that $\omega_{\mathrm{st}} = \pi/T_{\mathrm{st}}$ and $\omega_{\mathrm{sw}} = \pi/T_{\mathrm{sw}}$. By keeping the swing frequency ω_{sw} constant, then we have the stance frequency as follows

$$\omega_{\mathrm{st}} = (1-\beta)\omega_{\mathrm{sw}}/\beta, \tag{17}$$

which is a function of β. Moreover, substituting Eq. (17) into Eq. (10), the actual gait frequency ω_i can be rewritten as

$$\omega_i = \omega_{\mathrm{sw}}\left[1 + \frac{1-2\beta}{\beta(1+\mathrm{e}^{bv_i})}\right], \tag{18}$$

which also is a function of β. It is obvious that $\omega_i = \omega_{\mathrm{sw}}$ when $\beta = 0.5$.

In order to modulate the weight factor k_{ij}, we first note that the gait pattern is relevant to the duty factor β. In trot gait, a quadruped robot has at least two legs which simultaneously contact ground at any instant during locomotion; this implies that the duty factor β satisfies $\beta \geqslant 0.5$. In walk gait, the robot has at least three legs which contact ground at each time, so it requires β to satisfy $\beta \geqslant \varpi$ with ϖ a proper threshold (here set as $\varpi \geqslant 0.85$). Then, we can define the alternation of gait patterns as

$$\boldsymbol{K} = \begin{cases} \boldsymbol{K}_{\mathrm{walk}}, & 0.85 \leqslant \beta < 1 \\ \boldsymbol{K}_{\mathrm{trot}}, & 0.5 \leqslant \beta < 0.85 \end{cases}. \tag{19}$$

Furthermore, it is obvious that the states of the oscillator Eq. (9) should converge to the origin when the robot's locomotion stops, which can be achieved by

choosing $\mu < 0$. On the other hand, we know that the robot's locomotion also stops when the duty factor β tends to 1. Therefore, we can choose the parameter μ as follows

$$\mu = -k_\mu \mathrm{sat}_\varepsilon(\beta - \delta), \tag{20}$$

where k_μ is a positive constant, $\delta \in (0.8, 1)$ is an upper threshold and set as $\delta = 0.9$ in this paper, and $\mathrm{sat}_\varepsilon(\cdot)$ represents a saturation function such that

$$\mathrm{sat}_\varepsilon(\xi) = \begin{cases} 1, & \xi \geqslant \varepsilon \\ \xi/\varepsilon, & -\varepsilon < \xi < \varepsilon \\ -1, & \xi \leqslant -\varepsilon \end{cases},$$

with ε a small positive constant.

3.2 Gait Adaptation

Besides the ability of smooth gait switching, one of the most important features of legged robots is the ability of updating their gait parameters to suit the environment. In general, this adaptation can be achieved by the feedback via the CPG models and the robot kinematics.

The feedback added into the CPG model is given by the following dynamic model:

$$\left.\begin{aligned} \dot{c}_i &= k_{\mathrm{cp}}(\theta_{\mathrm{B}} - c_i) - k_{\mathrm{cd}}\dot{c}_i \\ S_i &= -k_{\mathrm{f}} c_i \end{aligned}\right\}, \tag{21}$$

where, k_{cp}, k_{cd} and k_{f} are control gains; c_i is an inner state. This is a basic sensory feedback term in all simulations of this study which makes the CPG entrained with an adjustable swing trajectory. In addition, one can see that this dynamic equation assures that the CPG output ζ_i is always smooth.

The feedback via the robot kinematics consists of two parts: an adjustment of the robot's posture and a modulation of the mapping functions. In general, a posture adjustment is needed to keep the robot's stability over uneven terrains. As mentioned in Ref. [6], when a robot moves on an inclined ground, the vestibulospinal reflex mechanism makes the downward-inclined legs extending and the upward-inclined legs flexing. Therefore, we choose the desired body pitch angle θ_{B} as

$$\theta_{\mathrm{B}}^{\mathrm{d}} = (1 - \kappa)\phi, \tag{22}$$

where $\kappa \in (0, 1)$ is a proportional coefficient, and ϕ is the local angle of the ground. This equation implies that the robot's body rotates not more than the terrain does.

Besides the above posture adjustment, the modulation of the mapping functions is also needed. For the supporting legs, when the robot moving on a single plane surface, the predictor-corrector type mapping function Eqs. (14) and (15) and the posture adjustment Eq. (22) compose a suitable gait generator. However, this generator maybe undesirable when the robot connecting two non-coincidence surfaces, because the angle ϕ used in Eq. (22) is uncertain and the desired heights of the robot's body are usually different on non-coincidence surfaces. To provide a smooth transition on two connecting different surfaces for the robot, we introduce here a virtual continuous trajectory for CG on irregular terrains, as shown in Fig. 4.

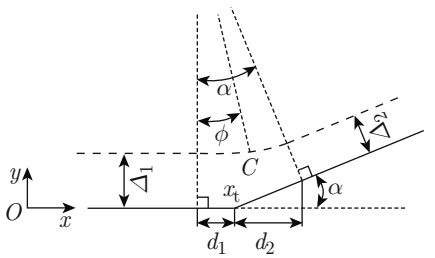


Fig. 4 Determination of the main body trajectory over transition terrains

In Fig. 4, the dashed line represents the virtual trajectory of CG; α is the angle of the slope; Δ_1 and Δ_2 denote the desired heights of the robot's body on the plain and the slope, respectively; x_{t} is the surface transition point's position along the x axis; d_1 and d_2 are the backward and forward distances of the transition corner, respectively. The continuity of this virtual trajectory is guaranteed by the weighting functions of ϕ and Δ, which are defined as

$$\phi = \begin{cases} 0, & x_{\mathrm{C}} < x_{\mathrm{t}} - d_1 \\ (x_{\mathrm{C}} - x_{\mathrm{t}} + d_1)^2 D\alpha, & x_{\mathrm{C}} \in [x_{\mathrm{t}} - d_1, x_{\mathrm{t}} + d_2] \\ \alpha, & x_{\mathrm{C}} > x_{\mathrm{t}} + d_2 \end{cases}, \tag{23}$$

$$\Delta = \Delta_1 + (\Delta_2 - \Delta_1)\phi/\alpha,$$

with $D = 1/(d_1 + d_2)^2$. Then, according to Eq. (22), we can choose the predicted trajectories for the supporting legs by

$$\boldsymbol{r}_{i,\mathrm{st}}^{\mathrm{d}} = \boldsymbol{J}(\phi) \begin{bmatrix} x_{\mathrm{f}i} - x_{\mathrm{h}i}^{\mathrm{d}} \\ y_{\mathrm{f}i} - y_{\mathrm{h}i}^{\mathrm{d}} \end{bmatrix} = \begin{bmatrix} k_x \zeta_i \\ -\left[\Delta \pm \dfrac{L_{\mathrm{B}}}{2}\sin(\kappa\phi)\right] \end{bmatrix}, \quad \forall v_i < 0. \tag{24}$$

Then, using Eqs. (14) and (15), we can consequently determine the corrected trajectories of the supporting legs. In addition, it should be noted that the robot must keep the values of Δ_1 and Δ_2 quite large to ensure its high mobility, and also usually has $\Delta_2 < \Delta_1$ to ensure its stability.

When a leg swings over the transition corner, the dragging or stumbling phenomenon should be avoided.

To this end, we design a virtual transition terrain at the corner. This virtual terrain is a flat rotating around the transition point, which coincides with the former terrain at the beginning and with the latter terrain at the end, respectively. The rotation completes in a single swing duration. Then, one can see that a leg can successfully swing over a convex corner ($\alpha < 0$), whereas it maybe drag over the concave corner ($\alpha > 0$). In order to resolve this problem, and from the definition of the angle ψ_i, we use it as a control signal and modulate the mapping function for a swinging leg Eq. (12) as follows

$$\left.\begin{aligned} \dot{\psi}_i &= k_{\psi\mathrm{p}}(\alpha - \psi_i) - k_{\psi\mathrm{d}}\dot{\psi}_i \\ y_{\mathrm{h}i,\mathrm{B}} &= y_{\mathrm{h}i}\cos\psi_i - (x_{\mathrm{h}i} - x_{\mathrm{t}})\sin\psi_i - \\ &\quad k_{\psi}\max\{\dot{\psi}_i, 0\} \end{aligned}\right\}, \tag{25}$$

where $k_{\psi\mathrm{p}}$, $k_{\psi\mathrm{d}}$ and k_{ψ} are positive constants to be chosen.

4 Simulation Results

In this section, we carry out some simulations to illustrate the performance of the proposed gait generators. The parameters of the robot are $L_{\mathrm{B}} = 0.8\,\mathrm{m}$, $L_{\mathrm{h}} = L_{\mathrm{k}} = 0.5\,\mathrm{m}$, and $\omega_{\mathrm{sw}} = \dfrac{\pi}{3}\,\mathrm{rad/s}$. The mass of the upper and lower links is $m = 1\,\mathrm{kg}$. The mass of the main body is $m_{\mathrm{B}} = 2\,\mathrm{kg}$. The initial joint angles are chosen as $\theta_{\mathrm{B}}(0) = 0$, $\theta_{\mathrm{h}i}(0) = -\dfrac{5\pi}{12}\,\mathrm{rad}$ and $\theta_{\mathrm{k}i}(0) = -\dfrac{\pi}{2}\,\mathrm{rad}$, $i = 1, 2, 3, 4$. The values of the CPG parameters are selected as $A_u = 5$, $A_v = 50$ and $b = 100$, whereas the initial values of the inner states are given by $\{u_1(0), u_2(0), u_3(0), u_4(0)\} = \{0, 0, 0, 0\}$ and $\{v_1(0), v_2(0), v_3(0), v_4(0)\} = \{0, -0.1, -0.03, 0\}$. The gait parameters are $k_x = 0.25$ and $k_y = 0.15$. The controller gains are set to $K_\theta = 100$ and $D_\theta = 50$. The spring and damper constants in Eq. (6) are given by $\{K_x, K_y\} = \{10^3, 10^4\}$ and $\{D_x, D_y\} = \{100, 500\}$, respectively. The parameters of the basic sensory feedback term Eq. (21) are $k_{\mathrm{cp}} = 10$, $k_{\mathrm{cd}} = 0$, $k_{\mathrm{f}} = 0.6$ and $\{c_1(0), c_2(0), c_3(0), c_4(0)\} = \{0, 0, 0, 0\}$.

Simulation results are shown in Figs. 5—8. At first, we perform a simulation for the gait transition of the robot from a trot gait gradually to a walk gait, and finally to stop, whereas the terrain is flat. In such case, the desired body pitch angle is $\theta_{\mathrm{B}}^{\mathrm{d}} = 0$. The parameters used in Eq. (20) are chosen as $k_\mu = 1$, $\delta = 0.9$ and $\varepsilon = 0.02$. The body height is set to $\Delta_1 = 0.8\,\mathrm{m}$.

In Fig. 5, the top panel shows the time evolution of the stimulation signal μ with respect to the duty factor β. The CPG outputs are shown in the middle panel, whereas the gait pattern diagram is illustrated in the bottom panel. Figures 6—8 show the time evolutions of the robot states and corresponding tracking errors Δx_{C}, Δy_{C}, $\Delta\theta_{\mathrm{B}}$, ΔQ_{h} and $\Delta\theta_{\mathrm{k}}$. From the last two panels of Fig. 5, one can see that the robot performs the correct gait event consequences during the whole simulation, and that the legs' coordination is changed gradually. In addition, Fig. 9 shows some results' comparison on the mapping function Eq. (13) and the proposed predictor-corrector method Eqs. (14) and (15). It can be seen that, the trajectories of CG and the body's angle are same in trot gaits using both methods, while they are different in walk gaits, and the responses of the body's pitch angle using the predictor-corrector method are more desirable than that using the mapping function Eq. (13).

To investigate the performance of the proposed gait adaptation scheme, a simulation is made for the robot moving on an irregular terrain with a trot gait. The test terrain is shown in Fig. 10, which consists of three

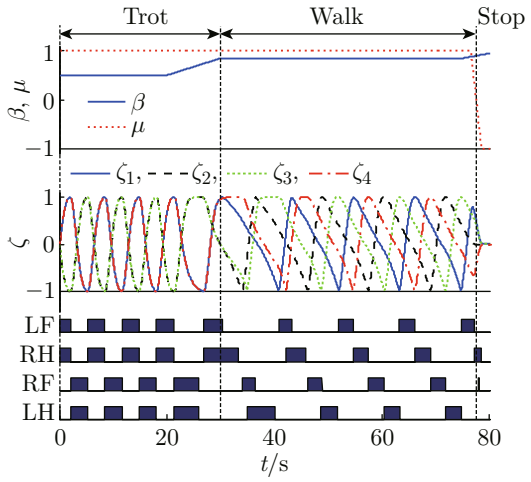


Fig. 5 Gait transitions via the duty factor β

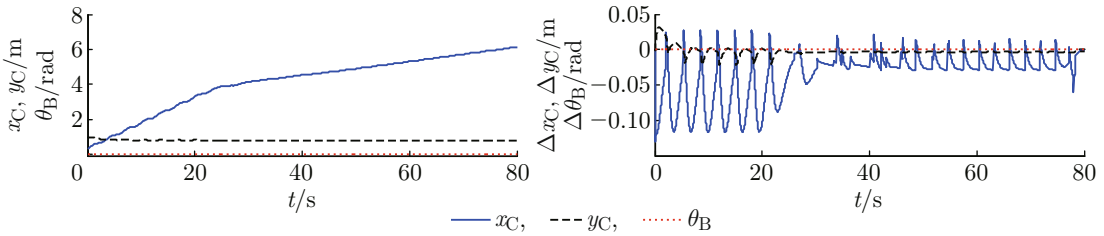


Fig. 6 Time responses of x_{C}, y_{C} and θ_{B} during gait transitions

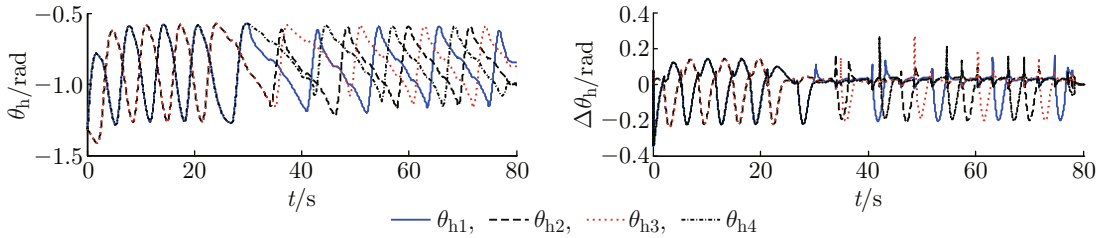


Fig. 7 Time responses of the hip joint angles during gait transitions

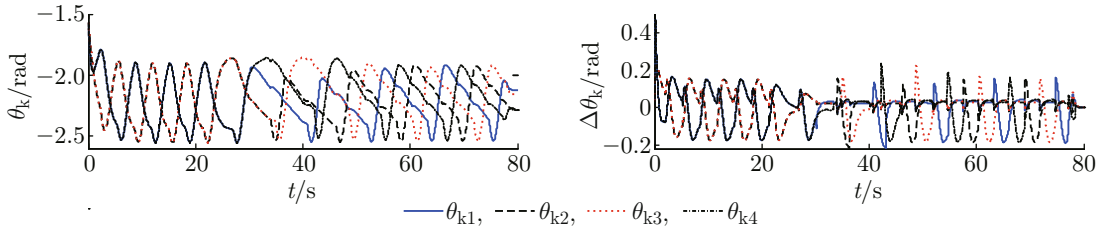


Fig. 8 Time responses of the knee joint angles during gait transitions

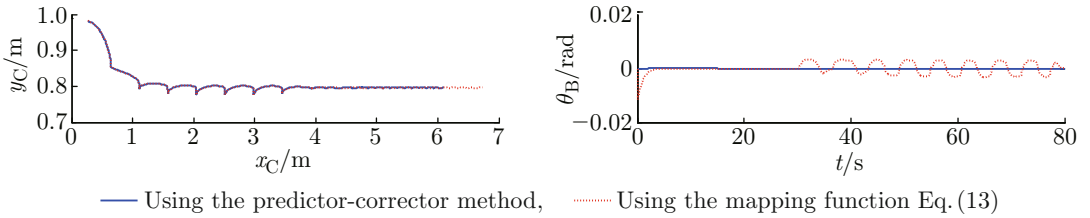


Fig. 9 Trajectories of CG and body's angle

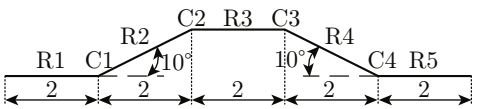


Fig. 10 Test terrain with uphill, flat and downhill components (m)

plain surfaces (i.e. R1, R3 and R5) and two 10° slopes (R2 and R4). It can be seen that the transition corners C1 and C4 are concave, whereas C2 and C3 are convex. The parameters of the gait and adaptation law are chosen as $\beta = 0.5$, $\omega_{\mathrm{sw}} = \dfrac{\pi}{2}$ rad/s, $k_x = 0.15$, $k_y = 0.1$, $d_1 = d_2 = L_{\mathrm{B}}/2$, $k_{\psi\mathrm{p}} = 2$, $k_{\psi\mathrm{d}} = 0$, $k_{\psi} = 0.15$ and $\kappa = 0.2$. The body height on slopes is set to $\Delta_2 = 0.75$ m.

The trajectories of the robot states and corresponding tracking errors are plotted in Fig. 11—13. From these figures, one can see that the robot can update its gait parameters to cope with the transition terrains. Figure 14 shows the time responses of CPG and the robot's configurations.

In order to illustrate the effects of the feedback term Eq. (21), we plot the time responses of the robot's stability margin in Fig. 15 with and without the effect of this term. The robot's stability margin is represented by a modified "wide stability margin (WSM)" introduced in Ref. [6], which is defined as the shortest distance from the projected point of CG to the supporting leg-ends. It can be seen that the WSM with the feedback term Eq. (21) is larger than that without the feedback term on the corners and slopes.

Furthermore, for the purpose illustrating the advantage of the proposed mapping function modulation Eqs. (23)—(25), we plot some comparison results for the robot (see Fig. 16). One can see that the trajectory

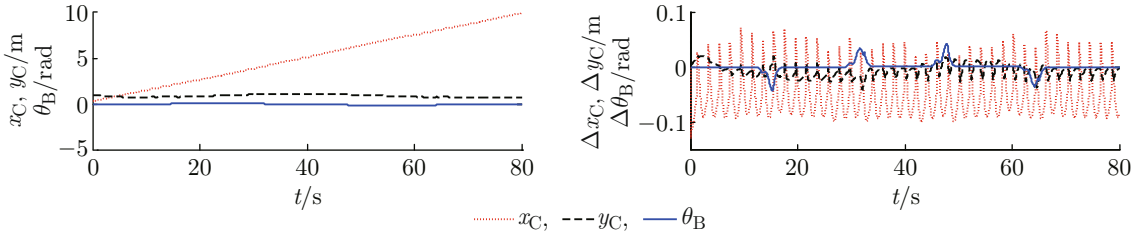


Fig. 11 Time responses of x_{C}, y_{C} and θ_{B} during gait adaptation

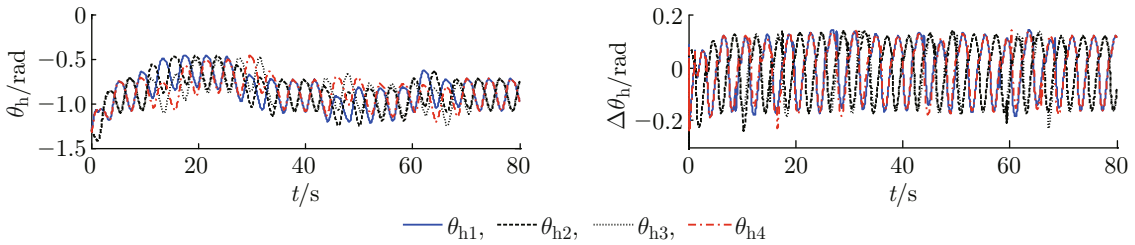


Fig. 12 Time responses of the hip joint angles during gait adaptation

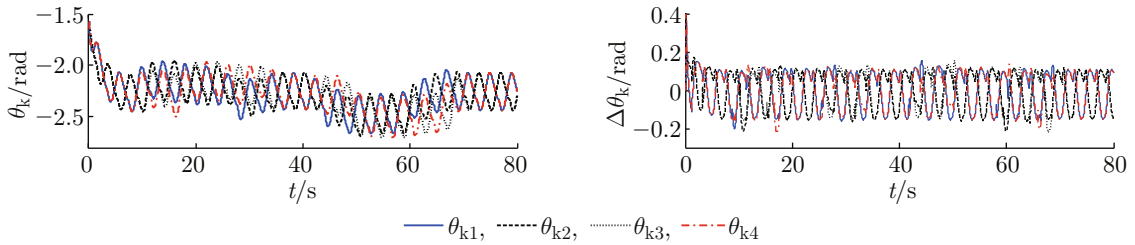


Fig. 13 Time responses of the knee joint angles during gait adaptation

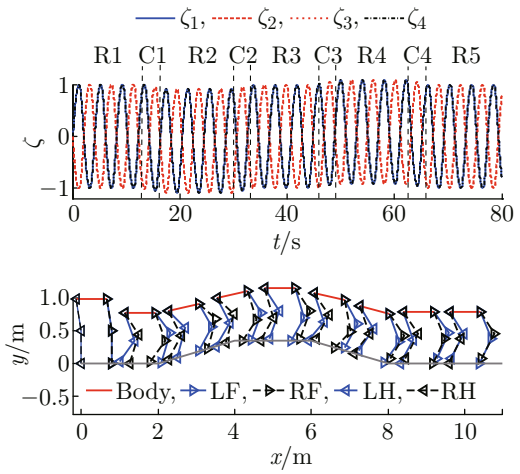


Fig. 14 CPG outputs and robot configurations

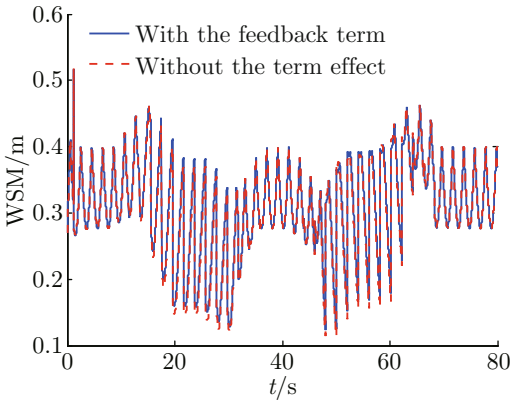
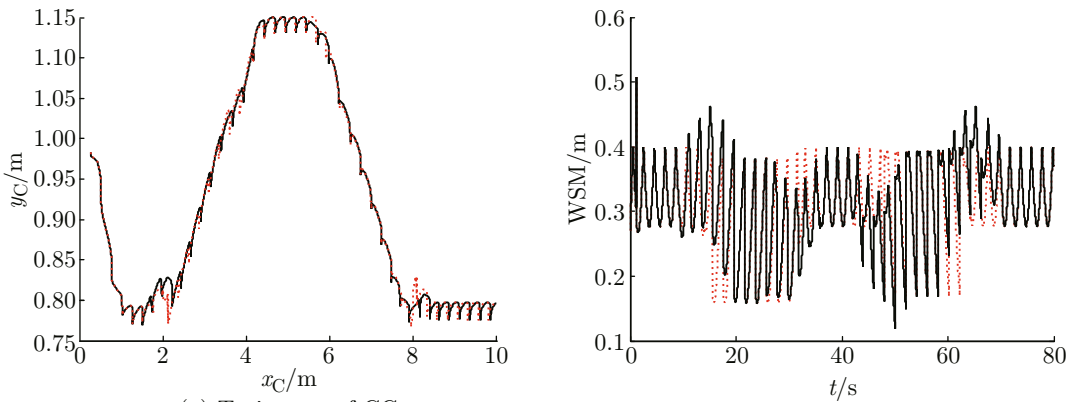


Fig. 15 Time evolution of WSM



(a) Trajectory of CG (b) Response of WSM

With the mapping function modulation, Without using modulation

Fig. 16 Comparison in terms of CG and WSM

of CG and the response of WSM over the corners by using the modulation Eqs. (23)—(25) vary more smoothly than that without the modulation.

5 Conclusion

In this paper, a CPG-based gait pattern controller is presented for a planar quadruped robot. The proposed locomotion generator consists of a unit CPG model and mapping functions. Using the duty factor as the control signal, a stimulation system is designed to realize smooth transitions of gait patterns. Moreover, the feedback via the CPG model and robot's kinematics is also designed. With the help of this modulation, an adaptive walking of the robot on irregular terrains can be realized. In the future, we will design more robust locomotion generators for legged robots to cope with the more nature and irregular terrains than that in this paper.

References

[1] Ijspeert A J. Central pattern generators for locomotion control in animals and robots: A review [J]. *Neural Networks*, 2008, **21**(4): 642-653.

[2] Matsuoka K. Sustained oscillations generated by mutually inhibiting neurons with adaptation [J]. *Biological Cybernetics*, 1985, **52**(6): 367-376.

[3] Liu C, Chen Q, Wang D. CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots [J]. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 2011, **41**(3): 867-880.

[4] Righetti L, Ijspeert A J. Pattern generators with sensory feedback for the control of quadruped locomotion [C]// *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*. USA: IEEE, 2008: 819-824.

[5] Santos C P, Matos V. Gait transition and modulation in a quadruped robot: A brainstem-like modulation approach [J]. *Robotics and Autonomous Systems*, 2011, **59**(9): 620-634.

[6] Fukuoka Y, Kimura H, Cohen A H. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts [J]. *The International Journal of Robotics Research*, 2003, **22**(3-4): 187-202.

[7] Liu C, Chen Y, Zhang J, et al. CPG driven locomotion control of quadruped robot [C]//*Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*. USA: IEEE, 2009: 2368-2373.

[8] Inagaki S, Yuasa H, Suzuki T, et al. Wave CPG model for autonomous decentralized multi-legged robot: Gait generation and walking speed control [J]. *Robotics and Autonomous Systems*, 2006, **54**(2): 118-126.

[9] Zhao W, Hu Y, Zhang L, et al. Design and CPG-based control of biomimetic robotic fish [J]. *IET Control Theory and Applications*, 2009, **3**(3): 281-293.

[10] Crespi A, Ijspeert A J. Online optimization of swimming and crawling in an amphibious snake robot [J]. *IEEE Transactions on Robotics*, 2008, **24**(1): 75-87.

[11] Wu X, Ma S. CPG-based control of serpentine locomotion of a snake-like robot [J]. *Mechatronics*, 2010, **20**(2): 326-334.

[12] Kamimura A, Kurokawa H, Yoshida E, et al. Automatic locomotion design and experiments for a modular robotic system [J]. *IEEE/ASME Transactions on Mechatronics*, 2005, **10**(3): 314-325.

[13] Ishiguro A, Fujii A, Hotz P E. Neuromodulated control of bipedal locomotion using a polymorphic CPG circuit [J]. *International Society for Adaptive Behavior*, 2003, **11**(1): 7-18.

[14] Taga G. A model of the neuron-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance [J]. *Biological Cybernetics*, 1998, **78**(1): 9-17.

[15] Kimura H, Fukuoka Y, Cohen A H. Biologically inspired adaptive dynamic walking of a quadruped robot [C]//*Proceedings of the 8th International Conference on the Simulation of Adaptive Behavior*. USA: MIT, 2004: 201-210.

[16] Matsubara T, Morimoto J, Nakanishi J, et al. Learning CPG-based biped locomotion with a policy gradient method [J]. *Robotics and Autonomous Systems*, 2006, **54**(11): 911-920.

[17] Asa K, Ishimura K, Wada M. Behavior transition between biped and quadruped walking by using bifurcation [J]. *Robotics and Autonomous Systems*, 2009, **57**(2): 155-160.