An adaptive locomotion controller for a hexapod robot: CPG, kinematics and force feedback

Article *in* Sciece China. Information Sciences · November 2014

DOI: 10.1007/s11432-014-5148-y

CITATIONS
26

READS
1,485

4 authors:

Weihai Chen
Beihang University (BUAA)
340 PUBLICATIONS **4,236** CITATIONS
SEE PROFILE

Guanjiao Ren
Beihang University (BUAA)
4 PUBLICATIONS **120** CITATIONS
SEE PROFILE

Jianhua Wang
Xi'an University of Technology
120 PUBLICATIONS **758** CITATIONS
SEE PROFILE

Dong Liu
Beihang University (BUAA)
16 PUBLICATIONS **283** CITATIONS
SEE PROFILE

SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2014, Vol. 57 112204:1–112204:18
doi: 10.1007/s11432-014-5148-y

An adaptive locomotion controller for a hexapod robot: CPG, kinematics and force feedback

CHEN WeiHai*, REN GuanJiao, WANG JianHua & LIU Dong

School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191*, China*

Received February 16, 2014; accepted April 18, 2014

Abstract Insects can perform versatile locomotion behaviors such as multiple gaits, adapting to different terrains, fast escaping, etc. However, most of the existing bio-inspired legged robots do not possess such walking ability, especially when they walk on irregular terrains. To tackle this challenge, a central pattern generator (CPG)-based locomotion control methodology is proposed, integrated with a contact force feedback function. In this approach, multiple gaits are produced by the CPG module. After passing through a post-processing circuit and a delay-line, the control signal is fed into six trajectory generators to generate predefined feet trajectories for the six legs. Then, force feedback is employed to adjust these trajectories so as to adapt the robot to rough terrains. Finally the regulated trajectories are sent to inverse kinematics modules such that the position control instructions are generated to control the actuators. In both simulations and real robot experiments, we consistently show that the robot can perform sophisticated walking patterns. What is more, the robot can use the force feedback mechanism to deal with the irregularity in rough terrain. With this mechanism, the stability and adaptability of the robot are enhanced. In conclusion, the CPG-base control is an effective approach for legged robots and the force feedback approach is able to improve walking ability of the robots, especially when they walk on irregular terrains.

Keywords hexapod robot, locomotion control, central pattern generator (CPG), kinematics, force feedback

Citation Chen W H, Ren G J, Wang J H, et al. An adaptive locomotion controller for a hexapod robot: CPG, kinematics and force feedback. Sci China Inf Sci, 2014, 57: 112204(18), doi: 10.1007/s11432-014-5148-y

1 Introduction

Legged locomotion is the most commonly used movement in the natural world. No matter human beings (bipedal locomotion), mammals (quadruped locomotion), or insects (hexapod locomotion), they all use legs to walk. Although some high-efficient transport machines have been invented, such as cars (wheeled) or tanks (tracked), their high-efficiency is only limited to relative flat terrains. Animals, especially insects, are expert in rough terrain walking [1]. However, the engineering counterpart of insects, bio-inspired legged robots, cannot yet mimic all the advantages of insects up to now. For enhancing the walking ability of legged robots, there are several problems to be solved, such as developing more powerful actuators [2,3] and exploiting new bio-inspired mechanisms [4]. In this article, we mainly focus on the bionic locomotion controller of robots.

*Corresponding author (email: whchenbuaa@126.com)

Generally speaking, there are three different control strategies [5]: model-based controller, reflexive controller and pattern-based controller.

Origin of thought of the model-based controller is coming from the controllers of industrial robots. Swing legs are treated as serial robots; support legs together with the body and ground are regarded as a parallel robot. The controller plans out every point for every step. Thus, kinematics and dynamics are calculated to obtain the control parameters (e.g., joint positions or torques). The typical examples of this control strategy are the "big dog" [6] and "little dog" projects [7,8]. There are also some other researches employing this method [9–11]. Advantage of this method is the precise trajectory control. However, drawbacks include high costing of computational resources and difficulty to scheme out different gaits. Also, for every step, the designer has to find proper contact points actively and then transfer the control signal into joint space.

The reflexive controller is mostly based on biological observation of the movement of real insects [12,13]. Reflexive controllers exploit sensory stimulus and response reactions to produce leg motion and gait coordination. The most well-known method is "Walknet", proposed by Cruse and his colleagues [14–16]. This method is a decentralized control strategy and it is easy to respond to sensory stimuli. Each leg has its own control center and all the units are coupled together according to some interactive rules. There is no predefined movement and every motion is motivated by sensory stimuli.

Another controller is the pattern-based controller, specifically the controllers inspired by the central pattern generator (CPG) originating from the vertebrate's spinal cord or invertebrate's ganglions [17]. This mechanism has been applied to different types of legged robots, such as the bipedal robot designed by Taga et al. [18] and Aoi et al. [19]; the quadruped robot by Kimura et al. [20] and by Zhang et al. [21,22]; and the hexapod robots by Arena et al. [23], Inagaki et al. [24] and Manoonpong et al. [25,26]. Other types of bio-inspired robots also employed the CPG-based mechanisms, such as the amphibious robots by Ijspeert et al. [27], snake-like robots by Crespi et al. [28] and Wu et al. [29], and the robot fish by Yu et al. [30,31]. It inherits lots of advantages of animals, such as distributed control, the ability to deal with redundancies, and fast control loops. This method can easily generate sophisticated gait patterns so as to be implemented on different robots. Other advantages are discussed in the review by Ijspeert [32].

In our previous work [33,34], we also employed similar CPG-based control for hexapod locomotion. It is, indeed, good at generating multiple gait patterns and easy to accomplish delicate locomotion of the robots. However, there are still some remaining problems to be solved. For example, as the physical walking machine is not perfectly symmetrical because of the assembly error, the robot may turn even when we send forward walking instructions. The linkages of robot would stretch since the pure CPG-based control does not contain any trajectory controller. When a leg moves leftwards, another leg may just move rightwards at the same time. Thereby it would cause the physical deterioration of the actuators with the passage of time. Also, the rough terrain walking performance is still not satisfactory, since usually there is no foot pressure feedback mechanism in a traditional CPG-based control.

To tackle these challenges, a new locomotion controller will be addressed in this article. The basic locomotion patterns (i.e., the different gaits) are produced by a CPG module, specifically, a chaotic CPG module. Dynamics of the central oscillation circuits are initially chaotic. When different control inputs are added to the circuits, the outputs can be controlled into different periodic orbits such that different gaits are generated. Then, the control signals are sent to a neural post-processing module to shape the signal into a proper range, and then into a delay-line to generate six oscillations for the corresponding six legs. After that, trajectory generators are introduced for each leg to obtain the desired feet trajectories. These predefined trajectories are regular and enough for relative flat terrains. To realize the rough terrain walking, a heuristic force threshold feedback method is proposed to adjust the foot trajectory. Finally, inverse kinematics is calculated so as to transfer the control signals into joint space and these signals are sent to the actuators (i.e., servo-motors).

The contribution of this article is that, we integrate kinematics calculation and force feedback into a CPG-based control method. In this way, this controller inherits the advantages of the model-based control, reflexive control and CPG-based control, respectively. (1) With the CPG oscillator, multiple gai-

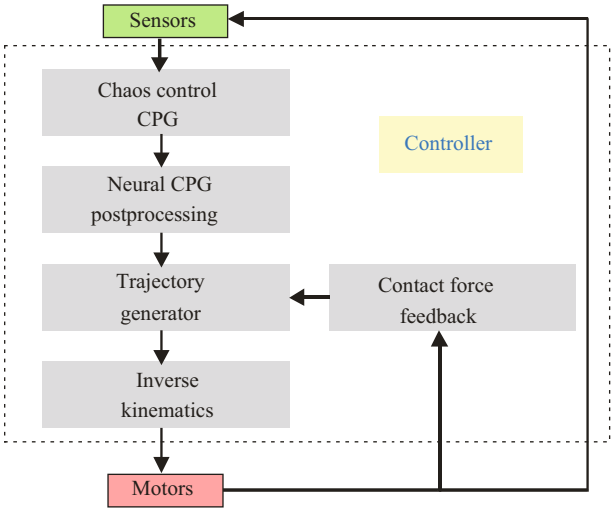

Figure 1 (Color online) The overall control structure of our hexapod robot. There are five modules including: (i) chaos control CPG module, (ii) neural CPG postprocessing module, (iii) trajectory generator module, (iv) inverse kinematics module and (v) contact force feedback module.

ts can be simply performed so that the robot can show different velocities and behaviors. These multiple gait patterns give the robot more freedoms to cope with different environmental stimuli. (2) The trajectory generating module and inverse kinematics module inherit the advantages of the model-based controller, which are able to control the foot trajectory precisely comparing with the pure CPG-based controller. This makes the robot more controllable and the feet slip much less than before. (3) The force feedback module substitutes functions of the reflexive controller to some extent. Only when the feet touch the ground firmly and stably, will the robot continue with its next support phase. Also, comparing with a model-based method, we do not need global dynamics for the whole body and legs to accomplish force feedback. It just regulates the feet trajectories independently according to the pressure information, which means that the force feedback is localized. Comparing with a pure CPG-based method, the kinematics information and force feedback ease the oscillation of the body, such that the robot walks more stably. Therefore, this algorithm can simply achieve adaptive walking of the hexapod robot not only in flat terrains but also in rough terrains under different gaits.

The rest of this article is organized as follows. Section 2 addresses the overall control structure of our algorithm. This section contains several subsections, which are the CPG module, the post-processing and delay-line module, the trajectory generator, the force-threshold-based feedback module and the inverse kinematics module. We will describe the five modules in detail and address how they interact with each other. Section 3 introduces the physical walking machine platform, which is inspired by real insects. Section 4 demonstrates the performance of our force-integrated CPG controller in simulations and real robot experiments. The simulations and experiments show the effectiveness of our proposed method. In Section 5, we discuss the remaining issues of this work and give a comparison to other relevant algorithms. Section 6 gives a conclusion finally.

2 Control strategy

The proposed locomotion controller includes five modules (see Figure 1). (1) The CPG module generates the basic patterns. (2) The postprocessing module is employed to transform the CPG signals into periodic oscillations and to delay the signals to produce oscillations for each leg. (3) Then, the oscillations are passed through trajectory generators so as to generate the foot-contact-point trajectories, i.e., the predefined trajectories. These trajectories are used for flat terrain walking. (4) To increase the stability and adaptability of walking on rough terrains, contact force feedbacks are introduced to the controller. The feet trajectories are adjusted according to the force information from contact feedback. (5) Finally, the trajectories are input into inverse kinematics modules to generate the position control signals in joint

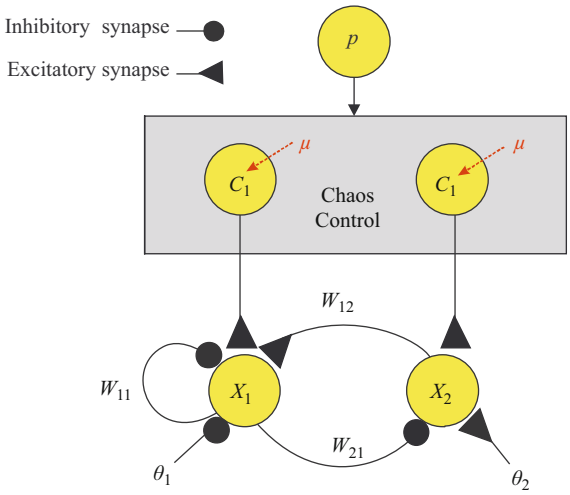

Figure 2 (Color online) The chaotic CPG controller. x_1 and x_2 indicate the neurons that generate the oscillation, while c_1 and c_2 are the control inputs depending only on the period p with a control strength μ. w_{11}, w_{12}, w_{21} represent the synaptic weights and θ_1 and θ_2 mean the biases.

space. Hence, the control instructions for all the actuators are produced.

CPG module is easy to generate different gait patterns, while kinematics module can control the foot trajectory precisely. Also, force feedback module increases the adaptability of the locomotion controller. Therefore, the hexapod robot can show versatile walking behaviors with this control structure. The following subsections will address functions of the five different parts in detail.

2.1 Chaos control CPG

The chaos control CPG unit is shown in Figure 2. Originally, the two neurons have self-connections as well as their mutual connections $(w_{11}, w_{12}, w_{21}, w_{22})$. They also oscillate spontaneously and generate a series of waves. To generate complex behavior of the hexapod robot, we modify the CPG structure by removing the self-connection of the second neuron. Using appropriate synaptic weights and biases, the CPG unit can exhibit chaotic dynamics. To achieve different walking patterns, we simultaneously add inputs to the two neurons, i.e., the control signals c_1 and c_2, which act as extra biases that depend only on the period p of the walking cycle. With an increase of p, the robot walks slower. The output of the neurons is detected every p steps and the chaotic dynamics is controlled to p-period orbit by adjusting the control input. The discrete time dynamics of the activity (output) states $x_i(t) \in [0, 1]$ of the circuit satisfies:

$$x_i(t+1) = \sigma(\theta_i + \sum_{j=1}^{2} w_{ij} x_j(t) + c_i^{(p)}(t)) \quad \text{for } i \in \{1, 2\}, \tag{1}$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$ is a sigmoid activation function with biases θ_i. w_{ij} is the synaptic weight from neuron j to i.

For a given period p, the control input

$$c_i^{(p)}(t) = \mu^{(p)}(t) \sum_{j=1}^{2} w_{ij} \Delta_j(t) \tag{2}$$

is calculated every p time steps while for the other steps it is set to 0. In Eq. (2), $\Delta_j(t)$ indicates the activity difference between the current step and p-steps before:

$$\Delta_j(t) = x_j(t) - x_j(t-p) \tag{3}$$

and $\mu^{(p)}(t)$ means the control strength, which changes its value adaptively according to:

$$\mu^{(p)}(t+1) = \mu^{(p)}(t) + \lambda \frac{\Delta_1^2(t) + \Delta_2^2(t)}{p} \tag{4}$$

with an adaption rate λ, e.g., 0.05.

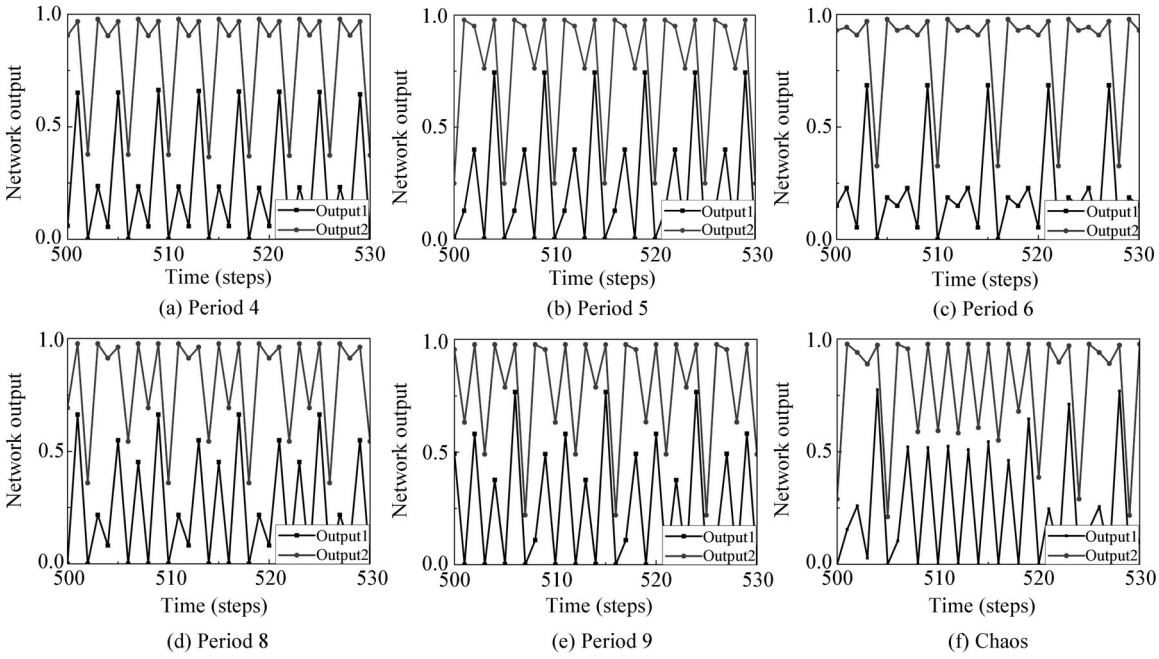

Figure 3 The network outputs of the CPG module. Panel (a)–(e) shows the CPG dynamics with control to specific periodic orbits, i.e., periods 4, 5, 6, 8 and 9, respectively. Panel (f) shows the CPG dynamics without control (chaotic).

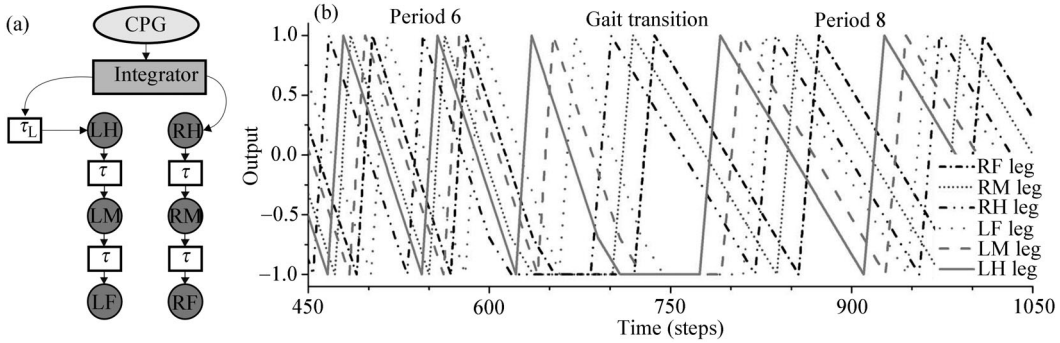

Figure 4 The working principle and outputs of the postprocessing module. (a) The lower circles represent the six legs, each of which contains a trajectory generation module and an inverse kinematics module (see Subsections 2.3 and 2.4). (b) The control signal is in the range from -1 to 1 with periodic ascending and descending. Ascending process indicates the swing phase, while descending process indicates the support phase. The CPG network oscillates at period 6 initially, and then it transfers to period 8.

Thus, the chaos control CPG module can produce different periodic patterns (see Figure 3, which shows the outputs of the two neurons). Without control input, the neural circuits show chaotic dynamics originally ($c_i^{(p)}(t) \equiv 0$, panel (f)). With inputting the control signal and adjusting the p value, the neural circuits can stably perform periodic patterns of periods 4–9 (shown from panels (a)–(e)), respectively. These five periodic patterns correspond to five commonly used gait patterns of the hexapod robot, which are tripod gait, tetrapod gait, transition gait, fast wave gait and slow wave gait, respectively. Note that, period 1 also exists, which means the robot stops (no oscillation). Periods 2, 3 and 7 are either not stable or not able to generate suitable gaits. Therefore we do not plot them here. When p is larger than 10 (included), we can also obtain stable periodic gait patterns but they are similar wave gait which walks slower. We can adjust walking velocity of the robot by simply changing the gait. With p increased, the walking velocity of the robot decreases. Here, chaos serves as a ground state of the CPG module.

2.2 CPG postprocessing module

The output from the CPG module is sent to a postprocessing module (see Figure 4) to format the oscillation wave into triangular shape and to produce six identical signals with phase lags for the six

legs. The output signal from the CPG circuit is firstly passed through a time window function so that the CPG signal is updated every $2p+1$ iteration steps. This subunit decreases the oscillation frequency to fit the implementation of motor control (the motor responses are not so fast as the initial oscillation frequency). Secondly, the signal is sent into a hysteresis unit for shaping the signal to binary values (-1 and 1). The hysteresis unit satisfies

$$a(t+1) = \tanh(w_{hys}a(t) + x_{in}(t)), \tag{5}$$

where $x_{in}(t)$ is the CPG output after the time window. $\tanh(x)$ is also a sigmoid function similar to the $\sigma(x)$ in Eq. (1). w_{hys} can be neither too big nor too small. If w_{hys} is too big, the output can hardly get out from the negative saturation status, and so it remains at -1. If w_{hys} is too small, the output is much sensitive to the input signal, and so the output cannot be stably saturate. It is empirically chosen. When an appropriate w_{hys} is set (e.g., 1.1 here), outputs can positively saturate ($+1$) or negatively saturate (-1). Thereby the binary values are generated. Thirdly, a signal integrator unit is employed to obtain continuous ascending and descending signals. In this unit, if the input is -1, the output value will increase more than the last step (swing phase). If the input is 1, the output value will decrease less than the last step (support phase).

After the integrator, the control signal is passed through a delay line to generate appropriate phase lags for different legs. From the biological observation [1] we know that, the recover sequence is from the rear leg to the middle leg then to the front leg. Therefore, the control signal is firstly sent to the RH leg. After the signal is delayed by τ steps and 2τ steps, it is implemented on the RM leg and the RF leg, respectively. For the left legs, before being sent to the LH leg, the control signal is first delayed τ_L steps. Then, it is delayed for τ steps for each leg, from rear to front (see Figure 4). These delays are independent of the target period or other influences. In our program, τ is 18 and τ_L is 90. Figure 4 shows the outputs of the six legs after the postprocessing module. The CPG unit oscillates initially at period 6 and then the oscillation transfers to period 8. The figure also indicates that the gait transition process is smooth and time-saving.

Note that, the CPG-generated triangular waves are different from the manually designed triangular waves. The CPG units can generate different gait patterns by simply adjusting the period p, without tuning other parameters. However, if we design the triangular waves manually, we have to plan out every gait and adjust the phase lag individually until it can generate the corresponding gait properly. Also, the gait transition could not be smooth since the manually designed waves do not have a convergent dynamics. Therefore, we cannot simply use manually designed triangular waves.

2.3 Trajectory generation

The conventional CPG-based strategies usually apply the post-processed control signal to some motor control modules, to convert the CPG output to the joint space. Such a method is a forward model. In contrast to the forward model, we use an inverse model in this article. In this way, the foot trajectory is under control such that movements of the legs do not stretch the robot. The CPG postprocessing outputs are fed into six trajectory generators to generate the feet trajectories. Here, we define three Cartesian Coordinate Systems for conveniently describing how to transmit the control signal into the feet trajectories (see Figure 5).

(1) *Inertial coordinate system* $O-xyz$, which is fixed on the ground. The origin of coordinates O is located at the center of mass at the original pose. The positive direction of y axis is the forward direction of the body, while the positive direction of z axis is opposite to gravity. Axis x is determined by the right-hand rule, i.e., x-positive points to the right sideward direction of the body.

(2) *Body coordinate system* $C-xyz$. The origin of coordinate C is located at the body's center of mass. The xCy plane is parallel to the body, and positive y has the same direction as the $O-xyz$ system. Also, the original position and orientation of $C-xyz$ is the same as $O-xyz$.

(3) *Leg coordinate system* B_i-xyz. The origin of coordinate B_i is at the center of the TC-joint, where the body connects to the relevant leg marked i ($i=1,2,\ldots,6$), and B_i is in the same plane with C in the body coordinate system.

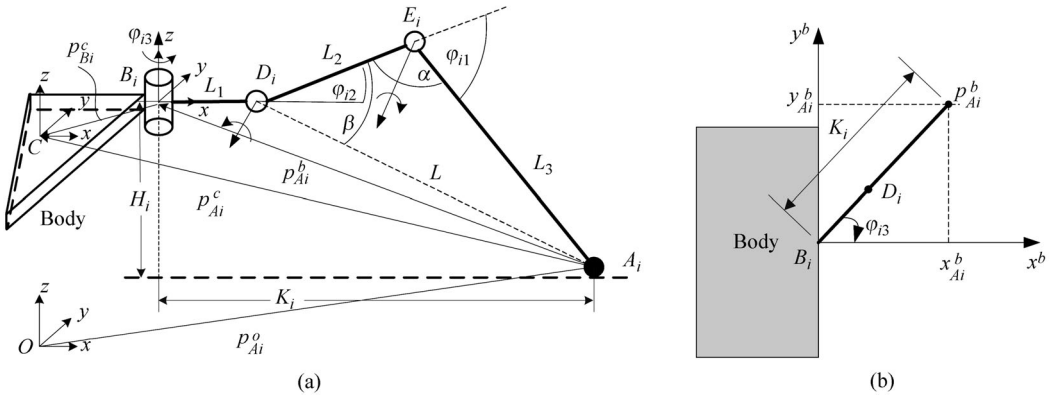

Figure 5 The kinematics model of the robot leg. The length of coxa, femur and tibia is L_1, L_2 and L_3, respectively. φ_{i3}, φ_{i2} and φ_{i1} indicate the angular value of TC-joint, CTr-joint and FTi-joint, respectively. (a) Front view; (b) top view.

We plan the feet trajectories in body coordinate system. For forward or backward walking, the x component of trajectory satisfies

$$traj_x = 0 \tag{6}$$

all the time. The robot can perform turning and sideward walking behaviors by adjusting $traj_x$, while these are beyond the scope of this article. In the forward direction (y direction),

$$traj_y = cpg_{post}, \tag{7}$$

where cpg_{post} means the output from the postprocessing module. In the swing phase, the leg moves from the posterior extreme position (PEP) to the anterior extreme position (AEP), i.e., the control signal ascends. In the support phase, the leg moves from AEP to PEP, i.e., the control signal descends. In the up and down direction (z direction), $traj_z$ satisfies

$$traj_z = \begin{cases} cpg_{post} + 1, & \text{in swing phase and } cpg_{post} \leqslant 0, \\ -cpg_{post} + 1, & \text{in swing phase and } cpg_{post} > 0, \\ 0, & \text{in support phase.} \end{cases} \tag{8}$$

Therefore, the feet trajectories are obtained. Note that these trajectories are the predefined trajectories, which can be adjusted through force feedback.

2.4 Inverse kinematics

To generate control instructions for the actuators, inverse kinematics is calculated to transform the trajectories into joint space.

Firstly, for support legs, the foot position and the body position satisfy:

$$p^o_{A_i} = p^o_c + R_c p^c_{A_i}, \tag{9}$$

where $p^o_c \in R^3, R_c \in SO(3)$ represent the position and orientation of the body center C under the inertial coordinate (see Figure 5(a)). $p^o_{A_i}, p^c_{A_i}$ stand for the position of contact point A_i under the inertial coordinate O and under the body coordinate C, respectively. Here, $i = 1, 2, \ldots, 6$ indicates different legs.

Eq. (9) can be transformed to:

$$p^c_{A_i} = R_c^{-1}(p^o_{A_i} - p^o_c). \tag{10}$$

On the other side,

$$p^c_{A_i} = p^c_{B_i} + p^b_{A_i}, \tag{11}$$

where $p^c_{B_i}$ stands for position of the rotation center of the TC-joint under the body coordinate. $p^b_{A_i}$ represents the position of the contact point A_i under the leg coordinate. $R_c \in SO(3)$ is a specially

orthogonal matrix, which satisfies $R_c^{-1} = R_c^{\mathrm{T}}$. Therefore, combined with Eq. (11), Eq. (10) can be modified to

$$p_{A_i}^b = R_c^T(p_{A_i}^o - p_c^o) - p_{B_i}^c. \tag{12}$$

In this equation, $p_{B_i}^c$ is determined by the mechanical structure. In the traditional model-based method, the foot position $p_{A_i}^o$ is known from the last swing phase, i.e., the place where the robot drops its leg. If the body position p_c^o and the body orientation R_c^T are known from the global planning, we can obtain the contact point position in the leg coordinate, i.e., $p_{A_i}^b$. However, in our CPG-based control, $p_{A_i}^b$ is only influenced by itself and the body status, without interacting with other legs, such that this is a decentralized control structure. The output of the trajectory generator can be directly used as the value of $p_{A_i}^b$. Therefore, the global $p_{A_i}^o$, p_c^o and R_c are unnecessary.

Secondly, for swing phase, the foot trajectory can be directly planed in the leg coordinate. Therefore, the inverse kinematics of support legs and swing legs can be attributed to one problem. The problem is to determine angles of the active joints $\varphi_{i1}, \varphi_{i2}, \varphi_{i3}$, if the trajectory of foot contact point under the leg coordinate ($p_{A_i}^b$) is known.

It can be seen from Figure 5(b) that the TC- joint meets

$$\varphi_{i3} = \arctan\left(\frac{y_{A_i}^b}{x_{A_i}^b}\right). \tag{13}$$

The center of the CTr-joint D_i under the leg coordinate $B_i - xyz$ can be expressed as (see Figure 5(a)):

$$p_{D_i}^b = [L_1 \cdot \cos(\varphi_{i3}), L_1 \cdot \sin(\varphi_{i3}), 0]^{\mathrm{T}}. \tag{14}$$

From geometrical relationship among $\triangle A_i D_i E_i$ and the linkages, φ_{i1} and φ_{i2} are obtained:

$$\varphi_{i1} = \alpha \pm \pi, \tag{15}$$

$$\varphi_{i2} = \beta \mp \arcsin\left(\frac{z_{A_i}^b - z_{D_i}^b}{L}\right), \tag{16}$$

where

$$\alpha = \arccos\left(\frac{L_2^2 + L_3^2 - L^2}{2L_2L_3}\right), \tag{17}$$

$$\beta = \arccos\left(\frac{L^2 + L_2^2 - L_3^2}{2L_2L}\right), \tag{18}$$

and $z_{A_i}^b$, $z_{D_i}^b$ represent the z axis component of the two points A_i and D_i under the leg coordinate. L_1, L_2 and L_3 indicate the length of linkage coxa, femur and tibia. L satisfies:

$$L = \sqrt{(x_{A_i}^b - x_{D_i}^b)^2 + (y_{A_i}^b - y_{D_i}^b)^2 + (z_{A_i}^b - z_{D_i}^b)^2}. \tag{19}$$

Therefore, the joint angle values of TC-joint, CTr-joint and FTi-joint (φ_{i3}, φ_{i2} and φ_{i1}) are obtained.

2.5 Contact force feedback

To increase the adaptability to rough terrain walking of the hexapod robot, a heuristic contact force feedback module is employed. Since the servo-motor is driven by position control instructions instead of torque control instructions, to realize contact force feedback, this module also plays the role of a force-position convertor. The contact point trajectory is regulated by the following rules. If the ground is lower than expected, the controller depressed the foot to reach the ground and maintain the foot depression till the end of the support phase. If the ground is higher than expected, the controller reduces the foot depression during the support phase so as not to lift the body too much. Using this approach, the feedback force can be attributed to several states leading to different control strategies.

The feedback force is divided by two force thresholds (see Figure 6). If the sustain force is larger than the high force threshold (F_H), this leg is in full support. If the support force is between the high force

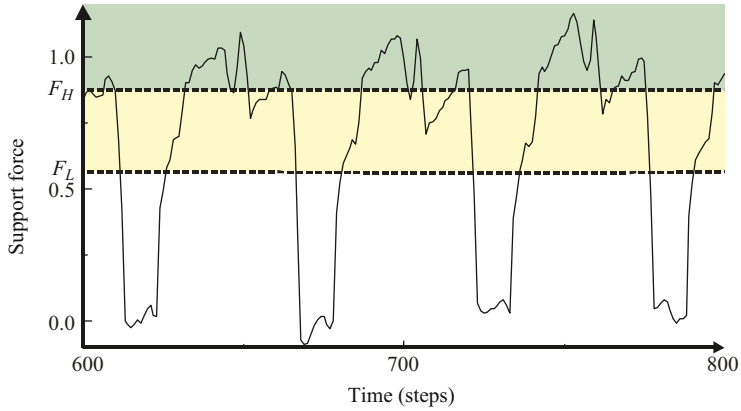

Figure 6 (Color online) The profile of the foot contact force. When the support force is larger than the high force threshold (F_H), this leg is defined in full support (light gray area). When the support force is less than the low force threshold (F_L), it is defined as no support (white area). When the support force is between F_H and F_L, it is called weak support (dark gray area).

No.	Foot contact force	Foot position	Control strategy
S1	$f \leqslant F_L$	Any (Support Phase)	Fast depress
S2	$F_L < f < F_H$	$d \geqslant D_{PRE}$	Maintain depression
S3	$f \geqslant F_H$	$d \geqslant D_{PRE}$	Slow elevate
S4	$f > F_L$	$d < D_{PRE}$	Slow depress

Figure 7 The four control strategies during the support phase (S1–S4). It is determined by two parameters, the foot contact force f and the current foot position d.

threshold (F_H) and the low force threshold (F_L), this leg is in weak support (this happens mostly when the leg transfers from the swing phase to the support phase). If the support force is less than the low force threshold (F_L), we define it as no support. This controller uses not only the foot support force information but also the current foot-end position as determinant of the control strategy. The foot trajectory is initially based on the predefined trajectory, i.e., the output from the trajectory generator. The predefined foot depression (D_{PRE}) is the maximum foot depression when walking on flat terrain while maintaining a particular body height and it is 0.0 in our program.

After clearly addressing these determinant factors of leg states, we can determine the four strategies of the force feedback mechanism (see Figure 7).

S1 This state is the beginning of the support phase and its main purpose is to depress the leg to the ground. If the foot has not reached the predefined position and the contact force is less than F_L, which indicates the leg has not fully touched the ground, the controller increases the foot depression at a high constant rate (e.g., z axis of the foot trajectory ($traj_z$) is decreased by 0.3 for every step). Also, in the later stages of the support phase, if the ground contact is lost because of loss of stability or slippery terrain, the controller will continuously try to make ground contact by depressing the leg.

S2 If the foot depression is more than the predefined position D_{PRE} and the leg is in weak support, i.e., the contact force is between F_L and F_H, the controller will maintain the current foot position in y axis and continue the support phase. This state happens when the terrain is lower than expected and could be considered as a gap in the terrain affecting only this leg.

S3 If the foot depression is more than predefined D_{PRE} and the leg is in full support, i.e., the contact force is larger than F_H, the controller will slowly elevate the foot at a constant rate to lower the body height (e.g., $traj_z$ is increased by 0.1 for every step). When the terrain is lower than the expected level, it could either indicate a gap in the terrain affecting only one leg, or a lower area (such as a slope) that would ultimately affect all the other legs. If it is a gap, the foot would return to normal in the next walking period. If the terrain is lower, all the legs would depress separately so as to raise the body

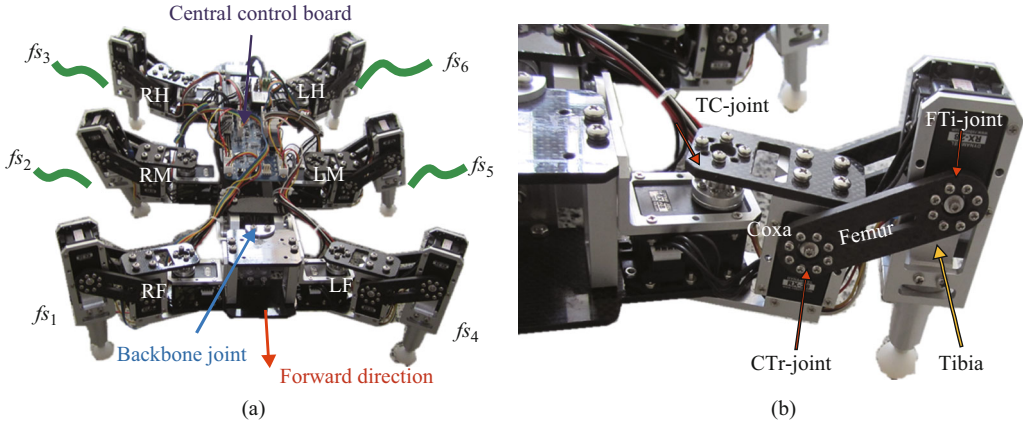

Figure 8 (Color online) (a) Biologically inspired hexapod robot. $fs_{1,2,\ldots,6}$ represent the force sensors. (b) Leg structure inspired by cockroach legs.

height. Therefore, the elevation rate must be low enough to maintain the locomotion stability and high enough to accomplish the desired body height.

S4 If the foot depression is less than predefined D_{PRE} (i.e., the terrain is higher than expected) and the contact force is larger than F_L, the control strategy is slowly increasing the foot depression (e.g., $traj_z$ is decreased by 0.1 for every step). The terrain could be higher because of a step or an obstacle. Thus, the controller could increase the body height so as to overcome the irregularity of terrain, specifically, the bump on terrain.

Note that, although the foot depression follows the four control strategies, limitations of the foot depression or elevation are set such that the leg would not exceed the mechanical constraint.

Above is the detail of our overall control structure. With this approach, different patterns can be generated by the CPG module and trajectory control is realized by the inverse kinematics module. For improving the rough terrain walking ability, force feedback is introduced. The experimental results section will demonstrate the advantages of this approach through simulations as well as real robot experiments.

3 Walking platform

A hexapod robot was developed to test our algorithm in real world experiments. Inspired by real insects [17], the robot has six identical legs. They are defined as right-front (RF) leg, right-middle (RM) leg, right-hind (RH) leg, left-front (LF) leg, left-middle (LM) leg and left-hind (LH) leg (see Figure 8(a)). Each leg contains three linkages, which can be considered as coxa, femur and tibia in the natural counterpart, insects, respectively. There are 18 degrees of freedom (DOFs) for leg motion, which means each leg has three rotation joints: the thoraco–coxal (TC-) joint enables forward and backward movements, the coxa–trochanteral (CTr-) joint enables elevation and depression of the leg, and the femur–tibia (FTi-) joint enables extension and flexion of the tibia, (see Figure 8(b)). The foot contact point is regarded as a 3-DOF spherical joint and it can support the leg in different orientations. Comparing to a real insect [17], the tarsus is ignored in the current design. In general, the tarsus is for absorbing outside impact forces and for sticking the leg to a walking surface [15]. Nevertheless, a spring system is installed in the leg to substitute part of the function of the tarsus, i.e., absorbing the impact force during touchdown on the ground. Specifically, force sensors are installed between the two springs (see Figure 9). In this way, the spring–sensor system can not only absorb the impact force, but also measure the sustain force for the purpose of contact feedback. The force sensors can also be used for recording and analyzing the walking patterns. With the force sensor installed between the two springs, it can move up and down during a walking period. In this way, spring 1 can prevent the sensor from being hit intermittently by the shaft. At the same time, spring 2 and the linear bearing ensure that only the vertical movement and force are transferred to the sensor. If we only use one spring, the force sensor might be damaged by the unceasing

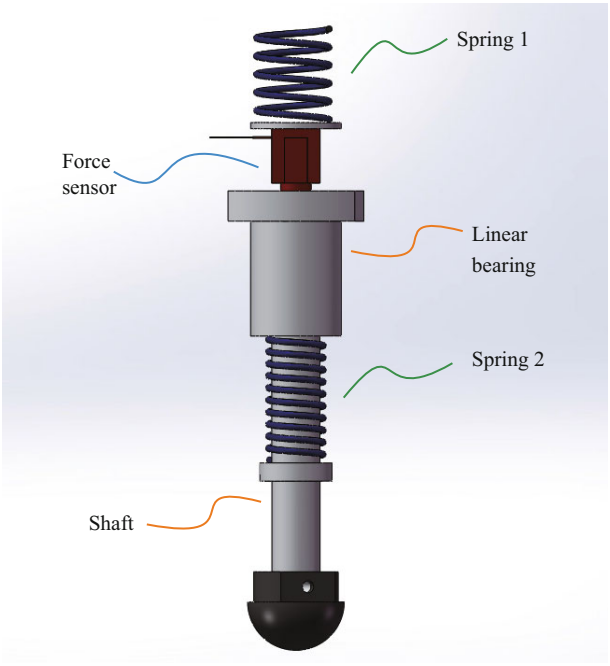

Figure 9 (Color online) The installation and working principle of the spring–sensor system.

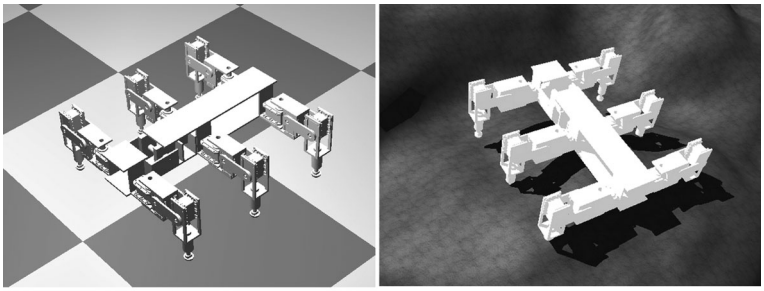

Figure 10 The hexapod robot in the simulation environment: (a) flat terrain and (b) rough terrain.

strike from the shaft. Therefore, spring 1 is installed on top of the sensor to counteract the shaft's movement.

Also, the robot contains two body parts, the front part and the hind part. The two body parts are connected by a backbone joint, which would be used for climbing experiments in future work but will not be discussed here. In total, the robot includes 19 active joints. Six infrared sensors are equipped at each leg, toward outside, for detecting obstacles near the corresponding legs. All the joints are driven by digital servo-motors, which are connected to a central control board via an RS485 interface. The central control board is based on an ARM (Cortex-M3, LPC1752) processor and all the sensory outputs are connected to it.

4 Experimental results

A simulation software, Webots^{TM}, was employed for testing our algorithm and collecting relevant data before it was implemented to the real walking platform. Figure 10 shows the simulation environment in both flat terrain and rough terrain. For the rough terrain, terrain irregularities were set randomly but did not exceed the lifting height of the leg, so that the robot was able to pass this terrain theoretically but needed regulating via force feedback.

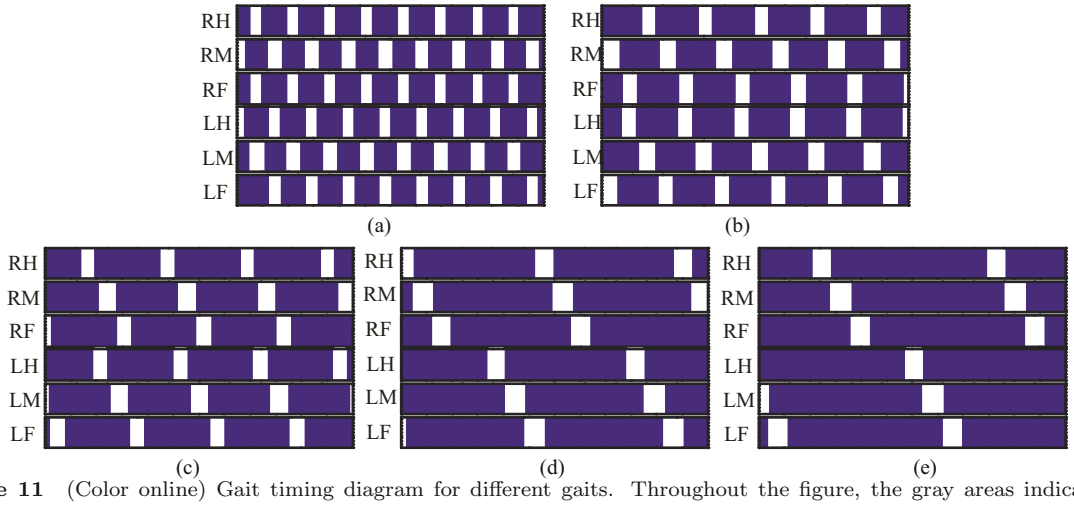

Figure 11 (Color online) Gait timing diagram for different gaits. Throughout the figure, the gray areas indicate the support phase (the contact force is larger than F_L) and the white areas indicate the swing phase (no foot contact). All the data were recorded from the force sensors in the simulation. (a) Period 4: Tripod gait; (b) Period 5: Tetrapod gait; (c) Period 6: Transition gait; (d) Period 8: Fast wave gait; (e) Period 9: Slow wave gait.

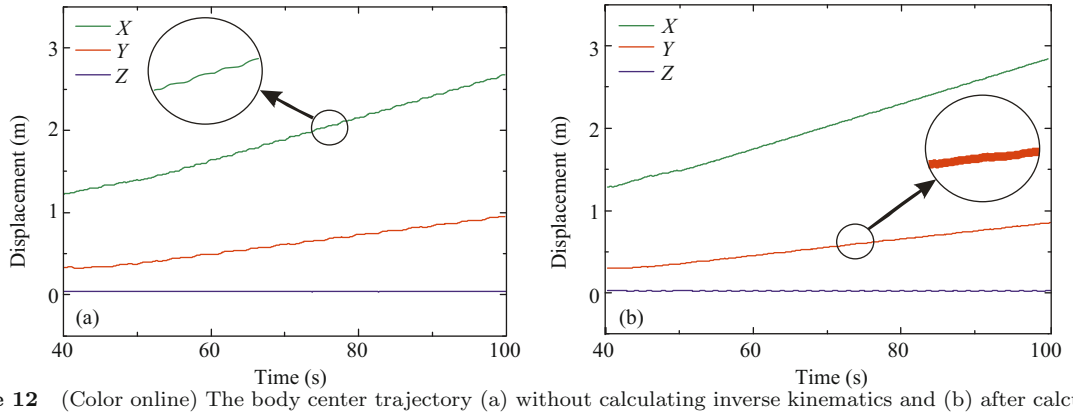

Figure 12 (Color online) The body center trajectory (a) without calculating inverse kinematics and (b) after calculating the inverse kinematics.

4.1 Flat terrain walking

The flat terrain walking was firstly tested to show the multiple walking patterns (different gaits). In all the simulations and experiments, the foot lifting height was set to 5 cm and the stride was set to 10 cm (i.e., the distance from the origin point to AEP and to PEP are both 5 cm). Therefore, the predefined foot trajectory was a semicircle.

The walking period p was set to 4, 5, 6, 8 and 9, respectively so that tripod gait, tetrapod gait, transition gait, fast wave gait and slow wave gait were performed, as shown in Figure 11. Generally speaking, with p increasing, the duty factor (time of swing phase divided by time of a whole walking period) decreased so that the robot walked gradually slower. If we set $p = 1$, the oscillation stopped and the robot did not move. Periods 2, 3 and 7 were either not stable patterns or not significant gait, and so we did not use them. When p was larger than 10, the robot showed slower and slower wave gait. We did not plot them here since they are similar to periods 8 and 9.

Most of the state-of-the-art CPG-based locomotion controllers were forward models, i.e., the motor control signals were generated directly from the CPG outputs without relating to mechanical parameters [24,25]. The transition from the CPG outputs to the motor control signals was always realized by passing through some neural networks, the parameters that were always adjusted empirically. Therefore, the foot trajectory could not be controlled precisely so that it might slip.

In contrast, our method is an inverse model, containing six trajectory generators and inverse kinematics modules. The feet trajectories can be controlled, such that the body moves smoother than before. For comparison, the effect of kinematics is shown in Figure 12. Without calculating kinematics, the center of

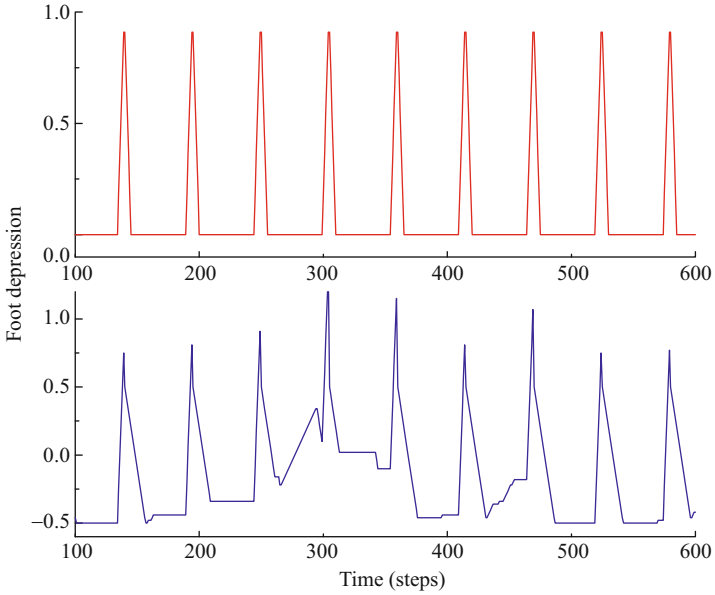

Figure 13 (Color online) The generated feet trajectories without support force feedback (upper) and with force feedback (lower). Data of both the two situations were recorded from the RF leg, when the robot walked in tetrapod gait (p5). In the upper panel, the unit of the vertical axis is normalized to 0 to 1. 1 indicates the highest point in swing phase and 0 indicates the predefined depression point. In the lower panel, the depression may exceed the range for adaptation.

the body's trajectory in X and Y direction would oscillate, see Figure 12(a). A part of the trajectory in the X direction is also zoomed in to see the detail of the oscillation. Although it appears to be okay, it is a large oscillation since the unit of displacement is meters. For real robot experiments, the oscillation would be amplified leading to uncontrollable walking because of the mechanical asymmetry. After passing through the kinematics module, the foot trajectory can be well controlled (see Figure 12(b)). Also, the robot walked much more straight. In simulation, the body center trajectory was nearly a straight line. In real robot experiments, the feet slipped much less such that the system might be easier to integrate other sensory information for possibly omnidirectional walking in future work. Moreover, it is easy to realize the contact force feedback, especially for irregular terrain walking. We only need to modify the predefined foot trajectory using the force information. Otherwise, it would be difficult to map the force information into joint space.

For the real robot experiment, please see supplementary video I. This video demonstrated the flat terrain walking experiment, in which the robot performed five different gaits with velocity increasing. The body center trajectory was nearly a straight line.

Supplementary video II shows the comparison of the walking performance without force feedback and with force feedback. This experiment was conducted in a flat terrain and the robot walked with tetrapod gait (period 5). Results showed that the walking performance was similar in the two conditions.

4.2 Rough terrain walking

The rough terrain walking was also tested to examine the effect of the proposed force feedback algorithm. The terrain was randomly set and the ground height ranged from 0 cm to 3.5 cm approximately. With this setting of ground height, on one hand, the body would not touch the ground to increase the friction. On the other hand, it was irregular enough to test our contact force feedback algorithm.

Figure 13 shows the foot contact point trajectory of the RF leg. The data were collected when the robot walked with period 5 (tetrapod gait). The upper panel indicates the trajectory without force feedback. It is a series of regular waves and it is normalized from 0 to 1. 1.0 means the predefined elevation height while 0.0 means the predefined depression height. The data is multiplied by the leg lifting height, i.e., 5 cm. The lower panel indicates the trajectory after the force feedback module. Based on the force feedback, the foot trajectory was adjusted according to the four control strategies described in Subsec-

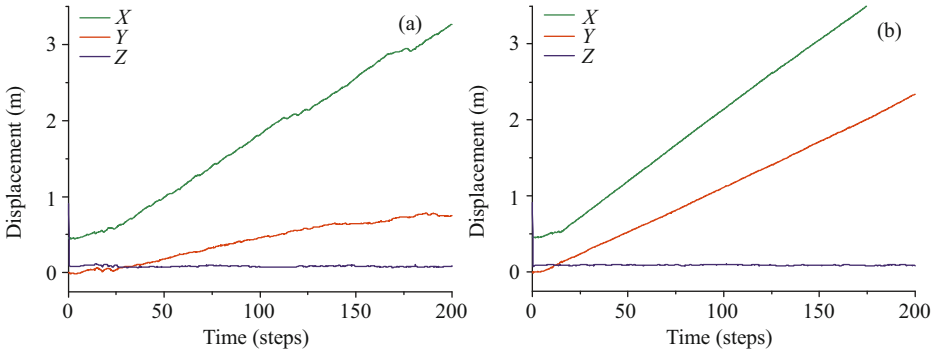

Figure 14 (Color online) The body center trajectory (a) before employing force feedback and (b) after employing force feedback.

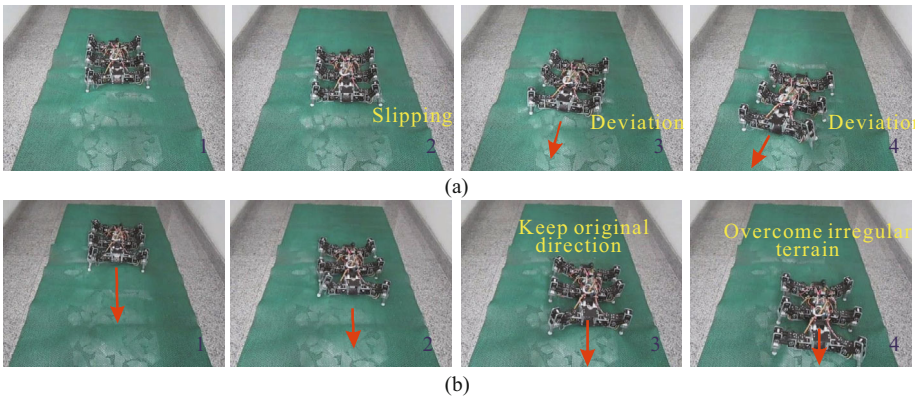

Figure 15 (Color online) Snapshots of the hexapod robot's walking on rough terrain. (a) Without force feedback; (b) with force feedback.

tion 2.5. The leg was elevated or depressed to adapt to different terrain texture. It can be seen from Figure 13 that the leg was sometimes depressed less than 0 to lift up the body to overcome the terrain irregularities. However, the minimum value was limited to -0.5 so that it would not exceed the mechanical constraints.

The effects of force feedback are shown in Figure 14, which depicts the body center trajectory in three dimensions. X and Y means the global coordinate parallel to the walking surface, and Z indicates the coordinate perpendicular to the ground. For comparison, panel (a) and panel (b) indicate the trajectories before implementing force feedback (directly using the predefined signals, e.g., the method in Refs. [25,26]) and after implementing force feedback (our method), respectively. Obviously, without force feedback, the robot could not keep walking straight so that the trajectory of body center was unsmooth and the robot would deviate from the desired direction. With contact force feedback, the foot depression was finely tuned so that the displacement of body center increased smoothly. Also, the body height in the vertical direction was kept nearly at a constant value from the ground. The performance of rough terrain walking with force feedback was similar to the performance of flat terrain walking.

Figure 15 shows the walking environment for the rough terrain. To construct a rough terrain, a plastic carpet was covered on the ground, under which polyfoam was filled. Therefore, the rough terrain was randomly set and the experiment condition was similar to the simulation. The rough terrain walking experiment was shown in supplementary video III. Without force feedback, the robot slipped on the ground and deviated from the original direction (see Figure 15(a)). With force feedback, the robot was able to adjust the leg depression and body height automatically so that it overcame the terrain irregularity easily (see Figure 15(b)). This video verified our proposed force feedback method.

For clearly showing the performance of force feedback and comparing with other non-force-integrated methods [25,26], we analyzed the orientation of the robot center. An Inertial Measurement Unit (IMU) was installed in the body to measure the roll–pitch–yaw (RPY) angles of the robot. Figure 16 shows the RPY angles of the robot before and after using force feedback mechanisms. With force feedback, the

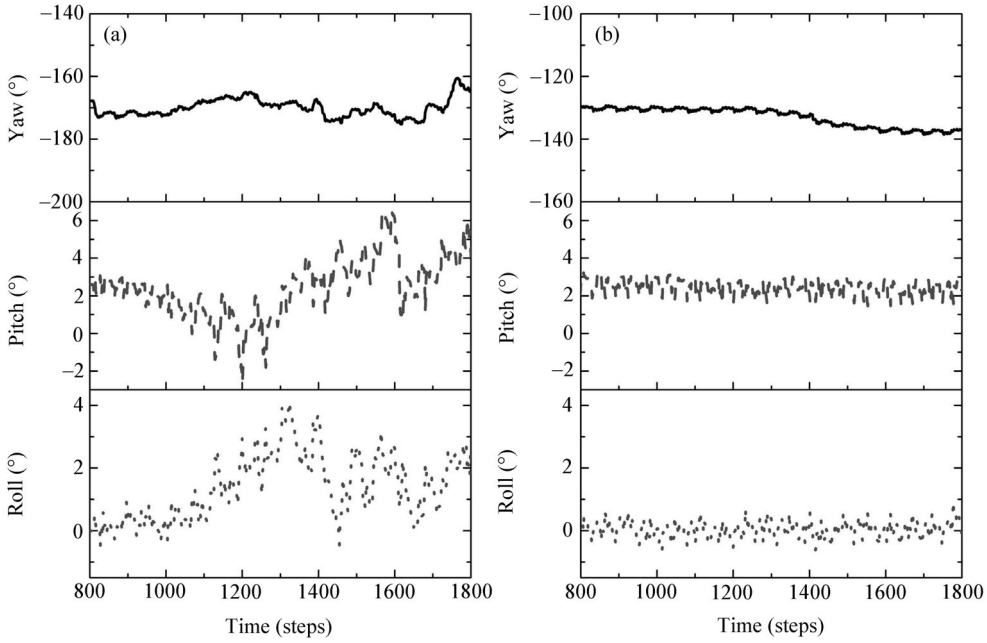

Figure 16 (Color online) The roll–pitch–yaw (RPY) angle of the robot body before implementing force feedback and after implementing force feedback. (a) Without force feedback; (b) with force feedback.

Figure 17 (Color online) Snapshots of the outdoor experiments.

magnitude of body oscillation during walking is decreased. Therefore, the robot is much easier to keep in its pre-defined behaviors. This figure also shows that the robot walks more stably than before.

4.3 Outdoor experiment

After testing the walking performance on flat and rough terrains, an outdoor experiment was conducted. The robot was initially placed at a granite pavement. As this terrain is stiff and regular, the robot walked quickly in tripod gait. After a period of time, the robot moved to gravel terrain, which is composed of little stones. Since this terrain was irregular and soft, the robot transferred to transition gait. After passing through the gravel terrain, the robot went through a long grass terrain. For increasing the velocity at the comparative long walking condition, the robot changed to tetrapod gait. At the end of the grass terrain, there was a slope at approximate 15° inclination. The robot used wave gait to climb up the slope, turn around and climb down the slope. Finally, the robot returned to a tile terrain and walked at tripod

gait again. The whole outdoor experiment lasted 10 minutes and the behaviors such as forward-walking, turning and side-walking were realized. The robots moved smooth and steady during the experiments. Supplementary video V shows this experiment and the snapshots are presented in Figure 17.

5 Discussion

This article addressed a locomotion control algorithm for hexapod robot, inspired by the central pattern generator in the neural system of insects. Several successful walking machines, which have been introduced in Section 1 [19,20,23,25–27], have also proved the effectiveness of similar methodology. The common points (also advantages) of these reports and our proposed algorithm include: the nonlinear oscillators show limit cycle behavior (i.e. stability), which can be used to resist perturbations; the robot can use multiple gaits to walk so as to perform flexible behaviors and so on. These are the advantages of the CPG-based controller compared with the traditional model-based methods. Buchli et al. [35] discussed different oscillators and analyzed the similarities and differences between them in detail. On the other side, the difference between our proposed method and the previous CPG-based methods is the implementation of the trajectory generator, the inverse kinematics module and the force feedback mechanism. The traditional CPG-based method is always a forward model, whereas our proposed approach is an inverse model. Although the sensory information does not directly influence the CPG dynamics such as in [20], the foot trajectory can respond to the sensors, specifically, the force sensors. In this way, each leg is only adjusted by its relevant force sensor, instead of taking the global model into consideration. Therefore, the control is decentralized.

In the viewpoint of stability margin (SM) [36], the center of gravity (CG) of the robot is located inside the support polygon, i.e., the SM is larger than zero. In the tripod gait, the support polygon is a triangle. In the other gaits, such as wave gait or tetrapod gait, the support polygon is much larger to hold CG. Hence, the locomotion is stable. For more information about stability analysis please refer to Ref. [36]. Also, after employing kinematics calculation and force feedback, the trajectories and the RPY angles of the robot body performs very little oscillation. The feet slip is much less than employing a forward model and the body behaves less left and right wiggle (see Figures 12, 14 and 16). Therefore, the robot walks more stably.

Also, this algorithm is easy to integrate force information. The input of the force feedback module is the pressure information from foot, while the output is a modulation signal (position control signal) for foot trajectory. In other words, the force feedback module is also a force–position convertor. In this way, it is easily implemented on our system although the servo-motors can only execute position control instructions. At the same time, this method avoids the complex dynamics calculation such as in [37–39]. It decouples the control of each leg such that the position control for each joint does not rely on the state of other legs and no multi-leg Jacobian is calculated. Leg control is completely localized-only a feedforward CPG-based trajectory signal is used to obtain joint positions. By decoupling the leg controllers, each leg can be handled as a distributed system that makes the robot control very convenient.

Although we did not address the issue of omnidirectional walking in this article (e.g., backward/sideward walking, turning, etc.), it will be discussed in future work. To deal with the terrain irregularity, we use the force feedback so that the Z-direction (vertical to the ground) predefined trajectory is adjusted. Similarly, if we use the ultrasonic sensor information and the infrared sensor information to adjust the X-direction (sideward direction) and Y-direction (forward direction) trajectories, the robot is able to turn and thereby the robot could achieve obstacle-avoiding behaviors.

6 Conclusion

In this article, we propose a CPG-based locomotion controller with trace tracking and force feedback functions. With this approach, the basic walking patterns is generated from chaos control of the CPG circuits. Thereby several different gaits are performed, such as tripod gait, tetrapod gait, transition gait

and wave gaits. The central oscillation is then sent into a postprocessing module and a delay-line so that six proper control signals are generated. Afterward, for each leg, the control signal is sent to an independent trajectory generator to produce the predefined foot trajectory. "Independent" here means every leg is controlled by only the output of CPG and sensory feedback from itself. Every leg does not interact with other legs, which make the control simple and decentralized. Then, force feedback modules are employed. With a threshold-based control, the contact force is evaluated and different adjusting strategies are implemented to regulate the predefined trajectory. Finally, the trajectory is input into an inverse kinematics module to generate the real-time position control instructions. The instructions are executed by the servo-motors such that the locomotion control is achieved.

A prototype, i.e. the bio-inspired hexapod robot, is applied to verify our control algorithm. It is tested in two situations: flat terrain and rough terrain. In both simulations and real robot experiments, we consistently show that the robot can follow the desired trajectory and perform sophisticated gaits. In rough terrain walking, the robot can overcome the terrain irregularity and walk better than that without force feedback. Therefore, the validity of our proposed algorithm is proved.

The contribution of this article is the integration of kinematics and force feedback into a CPG-base locomotion controller. Such mechanisms enhance the walking ability of the robot in not only flat terrain but also rough terrain. Our future work includes implementation of other sensors to show more intelligent behaviors, e.g., the omnidirectional walking.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61175108) and Innovation Foundation of BUAA for Ph.D. Graduates. We also acknowledge Dr. Poramate Manoonpong for technical advice on the chaotic controller and the robot configuration.

References

1 Wilson D. Insect walking. Ann Rev Entom, 1966, 11: 103–122
2 Kingsley D. A cockroach inspired robot with artificial muscles. Ph.D. Thesis. Case Western Reserve University, 2005
3 Ho T, Lee S. A fast mesoscale quadruped robot using piezocomposite actuators. Robotica, 2013, 31: 1–10
4 Quinn R, Offi J, Kingsley D, et al. Improved mobility through abstracted biological principles. In: Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, 2002. 2652–2657
5 Ferrell C. A comparison of three insect-inspired locomotion controllers. Robotics Auton Syst, 1995, 16: 135–159
6 Buehler M, Playter R, Raibert M. Robots step outside. In: International Symposium on Adaptive Motion of Animals and Machines, Ilmenau, 2005. 1–4
7 Kalakrishnan M, Buchli J, Pastor P, et al. Fast, robust quadruped locomotion over challenging terrain. In: Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Alaska, 2010. 2665–2670
8 Zucker M, Bagnell J, Atkeson C, et al. An optimization approach to rough terrain locomotion. In: Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA2010), Alaska, 2010. 3589–3595
9 Go Y, Yin X, Bowling A. Navigability of multi-legged robots. IEEE/ASME Trans Mechatr, 2006, 11: 1–8
10 Bai S, Low K, Teo M. Path generation of walking machines in 3D terrain. In: Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, 2002. 2216–2221
11 Silva M, Machado J. Kinematic and dynamic performance analysis of aitificial legged systems. Robotica, 2008, 26: 19–39
12 Dickinson M, Farley C, Full R, et al. How animals move: An integrative view. Science, 2000, 288: 100–106
13 Bläsing B, Cruse H. Stick insect locomotion in a complex environment: Climbing over large gaps. J Exper Biol, 2004, 207: 1273–1286
14 Schilling M, Cruse H, Arena P. Hexapod walking: An expansion to walknet dealing with leg amputations and force oscillations. Biol Cybern, 2007, 96: 323–340
15 Cruse H, Dürr V, Schilling M, et al. Principles of insect locomotion. In: Arena P, Patané L, eds. Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots. Springer, 2009. 43–96
16 Cruse H, Dürr V, Schmitz J. A bottom-up approach for cognitive control. In: Arena P, Patané L, eds. Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots. Springer, 2009. 179
17 Delcomyn F. Walking robots and the central and peripheral control of locomotion in insects. Auton Robots, 1999, 7: 259–270

18 Taga G, Yamaguchi Y, Shimizu H. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. Biol Cybern, 1991, 65: 147–159

19 Aoi S, Egi Y, Sugimoto R, et al. Functional roles of phase resetting in the gait transition of a biped robot from quadrupedal to bipedal locomotion. IEEE Trans Robotics, 2012, 28: 1–16

20 Kimura H, Fukuoka Y, Cohen A. Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. Inter J Robotics Res, 2007, 26: 475–490

21 Zhang X, Zheng H. Walking up and down hill with a biologically-inspired postural reflex in a quadrupedal robot. Auton Robots, 2008, 25: 15–24

22 Zhang X, Zheng H. Autonomously clearing obstacles using the biological flexor reflex in a quadrupedal robot. Robotica, 2008, 26: 1–7

23 Arena P, Fortuna L, Frasca M, et al. An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion. IEEE Trans Syst Man Cybern, Part B: Cybernetics, 2004, 34: 1823–1837

24 Inagaki S, Yuasa H, Suzuki T, et al. Wave CPG model for autonomous decentralized multi-legged robot: Gait generation and walking speed control. Robotics Auton Syst, 2006, 54: 118–126

25 Manoonpong P, Pasemann F, Wörgötter F. Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines. Robotics Auton Syst, 2008, 56: 265–288

26 Steingrube S, Timme M, Wörgötter F, et al. Self-organized adaptation of a simple neural circuit enables complex robot behaviour. Nature Phys, 2010, 6: 224–230

27 Ijspeert A, Crespi A, Ryczko D, et al. From swimming to walking with a salamander robot driven by a spinal cord model. Science, 2007, 315: 1416–1420

28 Crespi A, Ijspeert A. Online optimization of swimming and crawling in an amphibious snake robot. IEEE Trans Robotics, 2008, 24: 75–87

29 Wu X, Ma S. Adaptive creeping locomotion of a CPG-controlled snake-like robot to environment change. Auton Robots, 2010, 28: 283–294

30 Wu Z, Yu J, Tan M. CPG parameter search for a biomimetic robotic fish based on particle swarm optimization. In: Proceedings of 2012 IEEE International Conference on Robotics and Biomimetics, Guangzhou, 2012. 563–568

31 Yu J, Wei C. Towards development of a slider-crank centered self-propelled dolphin robot. Adv Robotics, 2013, 27: 1–7

32 Ijspeert A. Central pattern generators for locomotion control in animals and robots: A review. Neural Networks, 2008, 21: 642–653

33 Ren G, Chen W, Kolodziejski C, et al. Multiple chaotic central pattern generators for locomotion generation and leg damage compensation in a hexapod robot. In: Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012. 2756–2761

34 Chen W, Ren G, Zhang J, et al. Smooth transition between different gaits of a hexapod robot via a central pattern generators algorithm. J Intell Robotic Syst, 2012, 67: 255–270

35 Buchli J, Righetti L, Ijspeert A. Engineering entrainment and adaptation in limit cycle systems. Biol Cybern, 2006, 95: 645–664

36 Bai S, Low K, Zielinska T. Quadruped free gait generation based on the primary/secondary gait. Robotica, 1999, 17: 405–412

37 Ugurlu B, Kawasaki T, Kawanishi M, et al. Continuous and dynamically equilibrated one-legged running experiments: Motion generation and indirect force feedback control. In: Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012. 1846–1852

38 Boaventura T, Focchi M, Frigerio M, et al. On the role of load motion compensation in high-performance force control. In: Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012. 4066–4071

39 Li Y, Ahmed A, Sameoto D, et al. Abigaille II: Toward the development of a spider-inspired climbing robot. Robotica, 2012, 30: 79–89