Matsuoka's CPG With Desired Rhythmic Signals for Adaptive Walking of Humanoid Robots

Yong Wang, *Senior Member, IEEE*, Xihui Xue, and Baifan Chen, *Member, IEEE*

***Abstract*—The desired rhythmic signals for adaptive walking of humanoid robots should have proper frequencies, phases, and shapes. Matsuoka's central pattern generator (CPG) is able to generate rhythmic signals with reasonable frequencies and phases, and thus has been widely applied to control the movements of legged robots, such as walking of humanoid robots. However, it is difficult for this kind of CPG to generate rhythmic signals with desired shapes, which limits the adaptability of walking of humanoid robots in various environments. To address this issue, a new framework that can generate desired rhythmic signals for Matsuoka's CPG is presented. The proposed framework includes three main parts. First, feature processing is conducted to transform the Matsuoka's CPG outputs into a normalized limit cycle. Second, by combining the normalized limit cycle with robot feedback as the feature inputs and setting the required learning objective, the neural network (NN) learns to generate desired rhythmic signals. Finally, in order to ensure the continuity of the desired rhythmic signals, signal filtering is applied to the outputs of NN, with the aim of smoothing the discontinuous parts. Numerical experiments on the proposed framework suggest that it can not only generate a variety of rhythmic signals with desired shapes but also preserve the frequency and phase properties of Matsuoka's CPG. In addition, the proposed framework is embedded into a control system for adaptive omnidirectional walking of humanoid robot NAO. Extensive simulation and real experiments on this control system demonstrate that the proposed framework is able to generate desired rhythmic signals for adaptive walking of NAO on fixed and changing inclined surfaces. Furthermore, the comparison studies verify that the proposed framework can significantly improve the adaptability of NAO's walking compared with the other methods.**

***Index Terms*—Adaptive walking, central pattern generator (CPG), evolutionary algorithm, humanoid robots, neural network (NN).**

I. Introduction

OVER the last few decades, a variety of biologically inspired control methods has been proposed for controlling the movements of legged robots, most of which mimic the functions of central pattern generator (CPG) [1]–[3]. A CPG is a biological neural network (NN) which exists in invertebrate and vertebrate animals, and is responsible for the coordinated rhythmic movements of animals, such as breathing, chewing, and locomotion. A CPG generates coordinated multidimensional rhythmic signals through the mutual interconnection of a group of neurons. Studies conducted by neuroscientist have facilitated the design of neuron models for artificial CPG networks which attempt to imitate the behaviors of their biological counterparts. Matsuoka's CPG is a representative kind of CPG based on the neuron model proposed by Matsuoka [4]–[6], which has several attractive properties. First, it allows direct modulation of the frequencies and phases of the generated rhythmic signals by adjusting the values of some internal parameters and weights among neurons. In addition, it is also possible to make the generated rhythmic signals couple to the dynamics of mechanical systems automatically via rhythmic feedback, which is called the entrainment property [7]. During the coupling, the frequency of the generated rhythmic signals is synchronized with that of rhythmic feedback, and the final phase differences between the generated rhythmic signals and the rhythmic feedback are stable. Moreover, as demonstrated in [8], when the rhythmic feedback is large enough, it is possible to suppress the oscillation of the Matsuoka's CPG outputs, which shows faster entrainment and is helpful to maintain the balance of walking of humanoid robots.

Due to the above properties, Matsuoka's CPG has been extensively applied to legged robots. In 1991, Taga *et al.* [9] pioneered the use of Matsuoka's CPG for walking control of a simulated bipedal model. Williamson [7] exploited the entrainment property of a simple Matsuoka's CPG for arm joint control of a humanoid robot and achieved a range of rhythmic behaviors. Endo *et al.* [8] created the robust step motion of humanoid robot QRIO by allocating the Matsuoka's CPG neurons in the task space coordinate system and utilizing the entrainment property. Endo *et al.* [10] also took advantage of the entrainment property and applied a policy gradient

Manuscript received June 14, 2018; revised September 1, 2018; accepted September 9, 2018. This work was supported in part by the Innovation-Driven Plan in Central South University under Grant 2018CX010, in part by the National Natural Science Foundation of China under Grant 61673397 and Grant 61403423, in part by the Hunan Provincial Natural Science Fund for Distinguished Young Scholars under Grant 2016JJ1018, and in part by the Natural Science Foundation of Hunan Province of China under Grant 2018JJ3689. This paper was recommended by Associate Editor X. Zhu. *(Corresponding author: Baifan Chen.)*

Y. Wang is with the School of Information Science and Engineering, Central South University, Changsha 410083, China, and also with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: ywang@csu.edu.cn).

X. Xue and B. Chen are with the School of Information Science and Engineering, Central South University, Changsha 410083, China (e-mail: 0909110918@csu.edu.cn; chenbaifan@csu.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2870145

method to learn a feedback controller for Matsuoka's CPG, which enables QRIO to walk stably in both simulation and reality. The entrainment property is also successfully used for a quadrupedal robot Tekken to achieve adaptive walking on several irregular terrains [11]. In addition, Zhang and Zheng [12] and Liu *et al.* [13] made quadrupedal robots walk up and down hill adaptively by designing the proper control strategies based on Matsuoka's CPG combined with proper robot feedback. Researchers have manually adjusted the parameters of Matsuoka's CPG to obtain appropriate frequency and phase differences among the generated rhythmic signals [12], [13]. Liu *et al.* [14], [15] employed a control strategy to map the Matsuoka's CPG outputs to the trajectories of NAO's end effectors. Moreover, they introduced the pitch angle of NAO's upper body as robot feedback to make NAO adaptively ascend and descend a slope. Recently, with the development of numerical optimization techniques, the internal parameters and weights of the Matsuoka's CPG network can be tuned by making use of several powerful optimization algorithms such as evolutionary algorithms. As a result, the proper phase difference of multidimensional rhythmic signals can be obtained for versatile movements of humanoid robots [16]–[20].

From the above introduction, it can be seen that current studies focus mainly on the frequency and phase properties of Matsuoka's CPG when applying it to control the movements of legged robots. Note, however, that the precise adjustment of signal shape has been ignored unreasonably, which limits the adaptability of robot movement in various environments. Taking the walking of humanoid robots as an example, the shapes of the Matsuoka's CPG outputs should meet the required center of pressure (COP) trajectory to ensure that the humanoid robot can walk stably and adaptively on different inclined surfaces. However, it is impossible to precisely adjust the shapes of the Matsuoka's CPG outputs through simply modulating the internal parameters as well as weights among neurons, or introducing robot feedback. Therefore, how to obtain rhythmic signals with desired shapes is worthy of in-depth investigations.

To address this issue, we propose a new framework in this paper, which can generate desired rhythmic signals for Matsuoka's CPG. Overall, the main contributions of this paper can be summarized as follows.

1) In the proposed framework, we design three main parts: a) feature processing; b) NN learning; and c) signal filtering. Feature processing aims at transforming the Matsuoka's CPG outputs into a normalized limit cycle, which not only eliminates the shape information of the Matsuoka's CPG outputs but also preserves their temporal information. Afterward, NN learning generates desired rhythmic signals by learning from the feature inputs, which consist of the normalized limit cycle and robot feedback. Finally, signal filtering seeks to ensure the smoothness of the desired rhythmic signals.
2) The proposed framework is the first attempt to generate rhythmic signals with desired shapes while preserving the frequency and phase properties of Matsuoka's CPG. Moreover, it has several additional advantages, such as the ability to generate various desired rhythmic signals with high precision and the robustness against noise induced by the rhythmic feedback.
3) Based on the proposed framework, we develop a control system for adaptive omnidirectional walking of humanoid robot NAO on various inclined surfaces. In this control system, the rhythmic signals generated by the proposed framework are adopted as the rhythmic compensations for the Matsuoka's CPG outputs, two types of robot feedback are incorporated with the aim of enabling NAO to adapt itself to the changing environments, and a learning objective is defined for NN learning to generate desired rhythmic signals. Moreover, a popular evolutionary algorithm, called differential evolution (DE) [21], is applied to train the weights of NN.
4) The effectiveness of this control system is demonstrated by both the simulation and real experiments of NAO's walking on a fixed inclined surface and a changing inclined surface. Moreover, we compare the proposed framework with other methods to verify its superiority.

The rest of this paper is organized as follows. Section II introduces the preliminary knowledge regarding Matsuoka's CPG and its limit cycle. The details of the proposed framework are presented in Section III. Section IV reports the experimental results with the proposed framework. Section V describes the control system for adaptive omnidirectional walking of NAO and the corresponding experimental results are given in Section VI. Finally, Section VII concludes this paper.

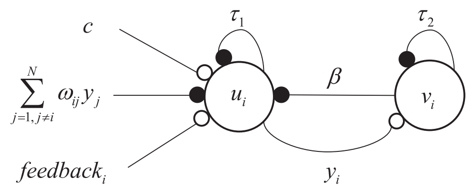


Fig. 1. Neuron model of Matsuoka's CPG. The lines ending with white and black dots indicate the excitatory and inhibitory connections, respectively.

II. Preliminary Knowledge

A. *Matsuoka's CPG*

The activity of a biological neuron is usually represented as a continuous-variable neuron model [22], [23]. Different from the commonly used neuron model [24], the neuron model proposed by Matsuoka takes the adaptation effect into account, which has been proven to play an important role in the generation of rhythmic signals [4]–[6]. Each neuron model in Matsuoka's CPG can be formulated as (1)–(3) and the corresponding structure is shown in Fig. 1

$$\tau_1\dot{u}_i = c - u_i - \beta v_i - \sum_{j=1,j\neq i}^{N}\omega_{ij}y_j + feedback_i \tag{1}$$

$$\tau_2\dot{v}_i = y_i - v_i \tag{2}$$

$$y_i = \max(0, u_i) \tag{3}$$

where u_i is the internal state of the ith neuron, v_i represents the degree of adaptation effect, y_i is the positive part of u_i, $\dot{u}_i$ and $\dot{v}_i$

TABLE I
PARAMETER SETTINGS OF MATSUOKA'S CPG

parameter	value
N	2
τ_1	1.47543
τ_2	1.8442875
c	2.0
β	4.01
ω_{12}, ω_{21}	3.0

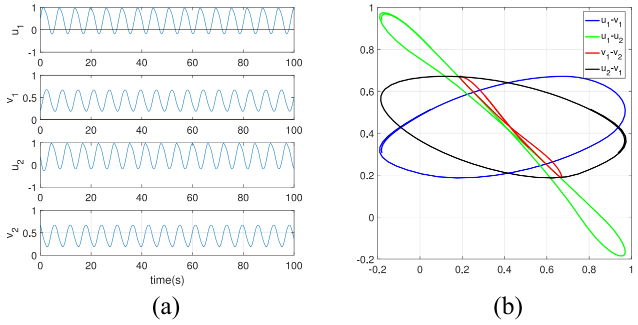


Fig. 2. Matsuoka's CPG outputs and the limit cycles in different phase planes. (a) Matsuoka's CPG outputs. (b) Limit cycles.

represent the differential of u_i and v_i, respectively, τ_1 and τ_2 determine the frequency of the Matsuoka's CPG outputs, the tonic excitation c modulates the amplitude of the Matsuoka's CPG outputs, β is the weight of adaptation effect, ω_{ij} is the connecting weight between the ith and jth neurons, N is the total number of neurons in Matsuoka's CPG, and $feedback_i$ is the rhythmic feedback from robots or external environments. In these equations, τ_1 and τ_2 also influence the shapes of the Matsuoka's CPG outputs, and $feedback_i$ is essential for the stable coupling between the Matsuoka's CPG system and mechanical systems.

B. Limit Cycle

In mathematics, the limit cycle is defined as a closed trajectory in a phase plane that arises in a dynamic system with stable oscillation outputs [25]. Matsuoka's CPG is a $2N$-dimensional dynamic system, which includes some tunable parameters as shown in (1)–(3). It is noteworthy that the Matsuoka's CPG outputs are sensitive to these parameters. With proper parameter settings, Matsuoka's CPG can generate stable rhythmic signals forming limit cycles [4]–[6]. In this paper, the parameters of Matsuoka's CPG were set as in Table I based on the experimental studies in Sections IV and VI. Here, we consider a Matsuoka's CPG with two neurons (i.e., $N = 2$) and suppose that $feedback_1 = feedback_2 = 0$. There are four state variables: u_1, u_1, v_2, and v_2. Accordingly, four rhythmic outputs are generated. As shown in Fig. 2(a), the four rhythmic outputs are stable since they oscillate continuously over time without attenuation or divergence. As a result, these outputs can form limit cycles. In Fig. 2(b), we plot four limit cycles, which are in the $u_1 - v_1$ plane, the $u_1 - u_2$ plane, the $v_1 - v_2$ plane, and the $u_2 - v_1$ plane, respectively.

III. PROPOSED FRAMEWORK

A. Overview of the Proposed Framework

Matsuoka's CPG, as formulated in (1)–(3), is a differential equation system. As a result, the Matsuoka's CPG outputs are coupled in time and space so that output shapes vary with the adjustment of frequency and phases. Therefore, once the frequency and phases of the Matsuoka's CPG outputs are specified, the shapes of these outputs are also determined. This relationship may drastically limit the adaptability of walking of humanoid robots when Matsuoka's CPG is applied for walking control. On the one hand, assuming that humanoid robots walk in a fixed environment, if the frequency and phases need to be adjusted, the new shapes resulting from such adjustment may not be suitable for the current environment and thus cause humanoid robots to fall down. On the other hand, assuming that humanoid robots walk in a changing environment, within the adjustable range of frequency and phases, there may not exist proper shapes for walking of humanoid robots due to the change of environment. Therefore, it poses a grand challenge for humanoid robots to adaptively walk in both fixed and changing environments.

Based on the above consideration, this paper proposes a new framework to regenerate the desired rhythmic signals whose shapes are not related to the frequency and phases of the Matsuoka's CPG outputs. In principle, each period of the rhythmic signal generated by Matsuoka's CPG is usually a continuous curve. It is well-known that with a sufficient number of hidden neurons, NN is able to fit any continuous curve in an interval via learning [26], [27]. Therefore, the proposed framework aims at applying NN to learn to regenerate desired rhythmic signals for Matsuoka's CPG.

To achieve this, three aspects should be considered.

1) Can the Matsuoka's CPG outputs be directly used as the feature inputs of NN? If the answer is no, then how can the Matsuoka's CPG outputs be processed?
2) What are the structure and learning objective of NN? How can the NN's weights be trained, and how can the desired rhythmic signals to suit the changing environments be generated?
3) Are the rhythmic signals generated by NN smooth when the frequency is changing? If the answer is no, then how can the discontinuous rhythmic signals be smoothed?

In our framework, three parts are designed to deal with the above three aspects, namely, feature processing, NN learning, and signal filtering. Fig. 3 depicts the proposed framework. Next, we will introduce each of these three parts.

B. Feature Processing

In fact, there exist two issues if we directly use the Matsuoka's CPG outputs as the feature inputs of NN. First, the Matsuoka's CPG outputs are expressed as points sampled on equal time intervals on the computer. However, these sample points are usually not uniformly distributed because of the irregular shapes of the Matsuoka's CPG outputs, which has a negative effect on the performance of NN learning to regenerate desired rhythmic signals. Second, due to the fact that the shapes of the Matsuoka's CPG outputs are related to frequency

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

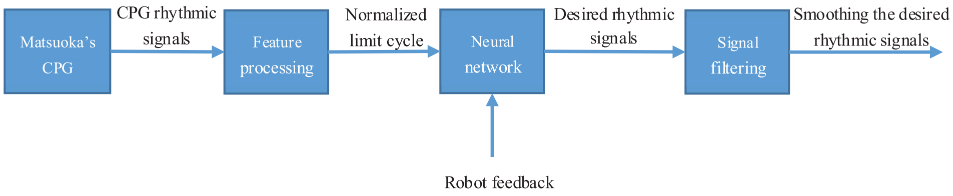


Fig. 3. Overview of the proposed framework.

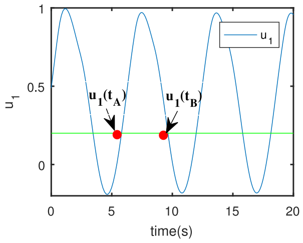


Fig. 4. Value of u_1.

and phases, the shapes of the rhythmic signals generated by NN will also be related to the frequency and phases of the Matsuoka's CPG outputs.

To address these two issues, the Matsuoka's CPG outputs must be processed before they become the inputs of NN, which is called feature processing in this paper. We denote the processed Matsuoka's CPG outputs as a feature vector $X(t) = (x_1(t), x_2(t), \ldots, x_k(t))$, where k is the dimension of feature vector. The elements of this feature vector are the NN's inputs. To enable NN to generate desired rhythmic signals at a specified frequency via learning, we consider that this feature vector should satisfy the following four conditions, where t denotes the time variable.

1) Each element of $X(t)$ should be a rhythmic signal.
2) The value of $X(t)$ should be different at different times in each period. That is, $\forall t_i, t_j \in [nT, (n+1)T)$ and $t_i \neq t_j$, $X(t_i) \neq X(t_j)$, where T denotes a period and n is a natural number.
3) In each period, the sample points of $X(t)$ with respect to the equal time intervals should be uniformly distributed in the feature space. This means that the distances between any two adjacent sample points in time are equal; thus, the variance of these distances tends to be zero, which can be expressed as

$$\frac{1}{m}\sum_{i=1}^{m}\left[(X(t_i)-X(t_{i+1}))^2-\frac{1}{m}\sum_{j=1}^{m}\left(X(t_j)-X(t_{j+1})\right)^2\right]^2 \to 0 \tag{4}$$

where m is the number of sample points in a period, and t_i and t_{i+1} are the ith time and the $(i+1)$th time, respectively.
4) The shape of the feature space formed by $X(t)$ should be fixed.

Next, we analyze the rationality of these four conditions. If we expect that the outputs of NN are rhythmic signals, the inputs must be rhythmic signals. Therefore, condition 1) can ensure that NN's outputs are rhythmic signals. If there exists a one-to-one mapping between the inputs and outputs, then all NN's outputs can be different in a period when all NN' inputs are different. In this way, any curve can be approximated by NN in a period. Accordingly, NN is able to approximate any desired rhythmic signal. Hence, condition 2) is necessary. In principle, the curve fitting performance of NN in an interval benefits from the uniformly distributed sample points. Thus, condition 3) is very useful for NN to generate the desired rhythmic signals. As pointed out, $X(t)$ is the processed Matsuoka's CPG outputs. If we make the shape of the feature space formed by $X(t)$ fixed, the shapes of the rhythmic signals generated by NN will not be influenced by the frequency and phases of the Matsuoka's CPG outputs. Therefore, condition 4) is also indispensable.

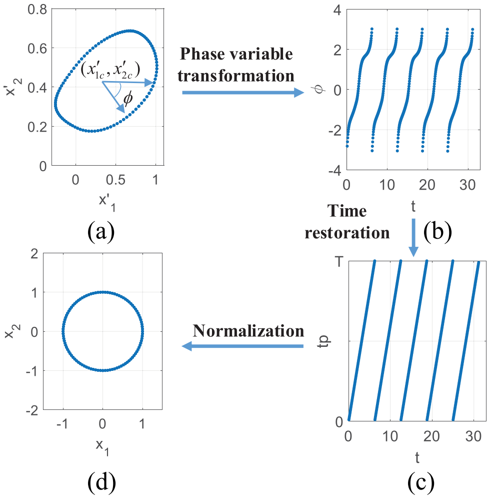


Fig. 5. Limit cycle normalization. In this figure, a blue point represents a sample point.

It is clear that if these four conditions could be satisfied, then the aforementioned two issues of the Matsuoka's CPG outputs would be addressed. To satisfy these four conditions, the feature processing proposed in this paper includes two steps: 1) feature selection and 2) limit cycle normalization. Specifically, the aim of the former is to satisfy conditions 1) and 2), and the aim of the latter is to satisfy conditions 3) and 4).

1) Feature Selection: First, we intend to select a feature vector from the Matsuoka's CPG outputs to satisfy conditions 1) and 2). As introduced in Section II-B, if Matsuoka's CPG is associated with proper parameter settings, all the outputs will satisfy condition 1) since they are rhythmic signals. Thus, the feature vector selected from the outputs also satisfies condition 1). Subsequently, we discuss how to select a feature vector satisfying condition 2). If only one output is selected,

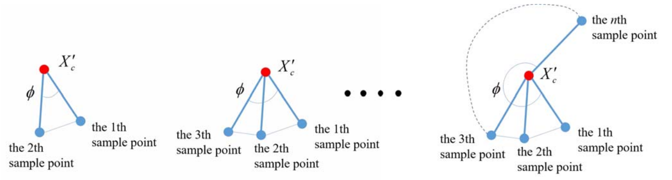


Fig. 6. Further explanation of ϕ in the limit cycle of Fig. 5(a).

for example u_1, it is impossible for the feature vector to satisfy condition 2). As illustrated in Fig. 4, due to the fact that u_1 is rhythmic, there must exist two moments in one period, for example t_A and t_B, to make $u_1(t_A) = u_1(t_B)$. As a result, the following issue will occur: if we expect that the outputs at t_A and t_B are different, NN cannot achieve this. In addition, if two outputs forming a limit cycle that crosses over itself are selected, such as u_1 and u_2 in Fig. 2(b), it still suffers from the above issue since the NN's outputs are the same at two moments of the intersection point. Therefore, it is necessary to select two outputs of Matsuoka's CPG which can form a limit cycle without crossing over itself, such as u_1 and v_1 in Fig. 2(b).

In this paper, the feature vector selected from the Matsuoka's CPG outputs is expressed as $X'(t) = (x_1'(t), x_2'(t))$, where $x_1'(t)$ and $x_2'(t)$ form a limit cycle without crossing over itself.

2) Limit Cycle Normalization: As shown in Fig. 2(b), the shapes of limit cycles formed by the Matsuoka's CPG outputs are usually irregular, and thus the sample points on these limit cycles are not uniformly distributed [Fig. 5(a)]. In addition, the shapes of the limit cycles are affected by the frequency and phases of the Matsuoka's CPG outputs. The above phenomena result in $X'(t)$ not meeting both conditions 3) and 4). Therefore, we need to eliminate the shape information in $X'(t)$ while preserving the temporal information. To achieve this, we propose a technique, called limit cycle normalization, to produce a normalized limit cycle. The advantages are twofold: 1) the sample points on the normalized limit cycle are uniformly distributed and 2) the shape of the normalized limit cycle is fixed. As shown in Fig. 5, limit cycle normalization is separated into three stages: 1) phase variable transformation; 2) time restoration; and 3) normalization.

In the first stage, $X'(t)$ is transformed into a 1-D phase variable ϕ. Fig. 5(a) and 5(b) show this transformation, which can be formulated as

$$\phi = a\tan 2\left(x_1'(t) - x_{1c}', x_2'(t) - x_{2c}'\right) \quad (5)$$

where $X_c' = (x_{1c}', x_{2c}')$ is an arbitrary point in the interior of the limit cycle $(x_1'(t), x_2'(t))$. Fig. 6 further explains ϕ in the limit cycle of Fig. 5(a). From Fig. 6, we can observe that the shape of a limit cycle is dependent on two factors. The first is the distances between X_c' and all sample points in a period, and the second is the distances between any two adjacent sample points in a period. In contrast, the change of ϕ from one time to another time depends only on the distance between the two adjacent sample points. As a consequence, by transforming $X'(t)$ into ϕ, partial shape information of the limit cycle has been excluded. Moreover, ϕ still contains the temporal information of a limit cycle.

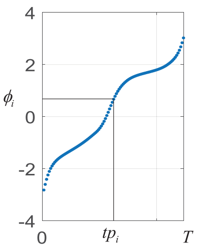


Fig. 7. Sampling process of ϕ in one period.

Next, we explain the second stage—the time restoration. For the sake of clarity, we put the sample points of one period in Fig. 5(b) into Fig. 7, where tp is the time variable in one period T. It is necessary to point out that the value of T is unknown, and thus tp cannot be determined. However, ϕ is known. Therefore, our aim is to restore tp by making use of ϕ. It is obvious from Fig. 7 that the distribution of tp is uniform, while the distribution of ϕ is not uniform. The mapping between ϕ and tp in Fig. 7 can be regarded as a sampling process [28]. Based on the sampling process in [28], we can obtain the following continuous expression from the probability distributions of ϕ and tp:

$$p(\phi) = p(tp)\left|\frac{dtp}{d\phi}\right| \Rightarrow tp = \frac{1}{p(tp)}\int_{-\infty}^{\phi} p\left(\hat{\phi}\right)\left|d\hat{\phi}\right|. \quad (6)$$

Then, (6) is discretized as follows:

$$tp_i = \frac{1}{p(tp_i)}\sum_{j=1}^{i} p(\phi_j)\left|\Delta\phi_j\right| \quad (7)$$

where $\Delta\phi_j = \phi_j - \phi_{j-1}$ and ϕ_j is the phase at tp_j.

$p(\phi_j)$ is defined as

$$p(\phi_j) = \frac{1/\left|\Delta\phi_j\right|}{\sum_{l=1}^{m-1} 1/|\Delta\phi_l|} \quad (8)$$

where m indicates the number of sample points in the limit cycle of one period and is recalculated if $\Delta\phi_k\Delta\phi_{k-1} < 0$ at the kth time.

Afterward, tp_i is expressed as follows:

$$tp_i = \frac{\sum_{j=1}^{i} 1/\left|\Delta\phi_j\right| \times \left|\Delta\phi_j\right|}{p(tp_i)\sum_{l=1}^{m-1} 1/|\Delta\phi_l|} = \frac{i}{p(tp_i)\sum_{l=1}^{m-1} 1/|\Delta\phi_l|} \quad (9)$$

where $0 \le i < m$, $p(tp_i)$ is a constant due to the uniform distribution of tp_i, and $\sum_{l=1}^{m-1} 1/|\Delta\phi_l|$ is also a constant since ϕ is known.

Thus, based on (9), tp_i can be easily restored by utilizing ϕ_i, and tp_i is uniformly distributed over the interval $[0, T]$ where $T = m/[p(tp_i)\sum_{l=1}^{m-1} 1/|\Delta\phi_l|]$.

Overall, compared with the first stage, all the shape information of the limit cycle has been excluded in the time restoration.

The last stage is to obtain the final feature vector that satisfies conditions 3) and 4), as shown in Fig. 5(d). Let

$$x_1(t_k) = \sin\left(\frac{2\pi\, tp_i}{T}\right) \quad (10)$$

$$x_2(t_k) = \cos\left(\frac{2\pi\, tp_i}{T}\right) \quad (11)$$

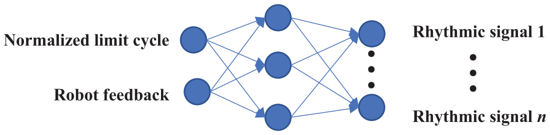


Fig. 8. Structure of NN.

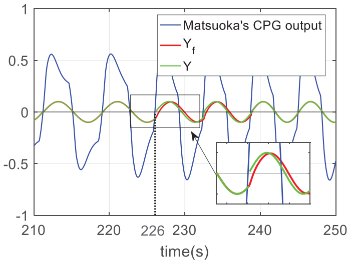


Fig. 9. Modified first-order low-pass filter for smoothing NN's output.

where t_k is the kth time, $t_k = (n-1)T + tp_i$, and n indicates the nth period. Due to the fact that $x_1^2(t_k) + x_2^2(t_k) = 1$, tp_i is projected onto the unit circle. The final feature vector is formulated as $X(t_k) = (x_1(t_k), x_2(t_k))$. Consequently, the irregular limit cycle formed by $X'(t)$ is transformed into a normalized one formed by $X(t_k)$. Furthermore, the sample points on this normalized limit cycle are uniformly distributed.

Remark 1: After feature selection and limit cycle normalization, the final feature vector $X(t_k)$ satisfies all the four conditions. Since the normalized limit cycle is the unit circle, its shape is always fixed regardless of the shape of the limit cycle formed by $X'(t)$. Additionally, it can be seen that the sample points on the normalized limit cycle and those on the limit cycle formed by $X'(t)$ has a one-to-one correspondence. Therefore, one can conclude that the normalized limit cycle discards all information about the shape of the Matsuoka's CPG outputs while preserving the temporal information. As a result, the shapes of the rhythmic signals generated by NN will not be affected by the frequency and phases of the Matsuoka's CPG outputs when $X(t_k)$ is used as the feature input of NN.

C. Neural Network and Learning

The structure of NN consists of three layers: 1) the input layer; 2) the hidden layer; and 3) the output layer. The adjacent layers are fully connected by weights, as shown in Fig. 8. For the input layer, in addition to the normalized limit cycle, robot feedback is added. The reason is because robot feedback is a reflection of the state of an environment; thus, it enables NN to learn to generate desired rhythmic signals for different environments. The output layer is responsible for generating rhythmic signals. The nodes of both the input and the output layers can be added flexibly. Therefore, it is easy to add robot feedback from various sensors to the input layer, and more than one desired rhythmic signal can be generated simultaneously in one network.

The shape of the rhythmic signal generated by NN is determined by NN's weights. Many weight training methods have been proposed for NN learning to fit a curve in an interval [29], [30], which are also applicable to our framework to generate desired rhythmic signals. However, different weight training methods suit different situations. When the desired rhythmic signals are known in advance, such as the imitation learning of walking control of humanoid robots, the required learning objective can be explicitly expressed as the difference between the desired rhythmic signals and the NN's outputs. Under this condition, the stochastic gradient decent method (SGD) [29] is a good choice for NN weight training. When the desired rhythmic signals are unknown, such as adaptive walking of humanoid robots on different surfaces, the required learning objective can only be implicitly expressed as some stability criteria. Under this condition, an evolutionary algorithm is a preferred method. Therefore, it is necessary to choose an appropriate weight training method for NN learning to generate desired rhythmic signals according to the characteristics of the situations.

D. Signal Filtering

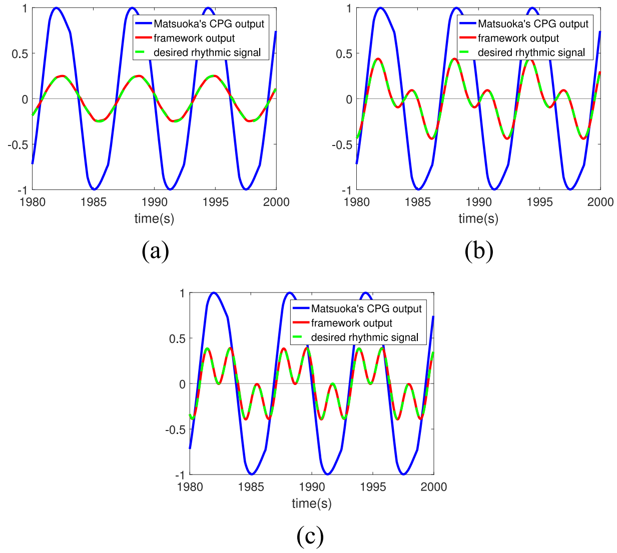
Feature processing ensures that NN's outputs are continuous if the frequency of the Matsuoka's CPG outputs is fixed, which is beneficial for smooth control of the robot's trajectory. However, if we need to change the frequency of the Matsuoka's CPG outputs to suit the dynamics of the robot's mechanical system (entrainment property) after learning, the NN's outputs will be discontinuous due to the change in the number of sample points (i.e., m) in each period. For example, as shown in Fig. 9, the desired rhythmic signal is a sine-like signal. After NN learning, NN generates rhythmic signal Y (shown in the green line). When the frequency of the Matsuoka's CPG output (shown in the blue line) increases at 226 s, Y becomes discontinuous during the transition between two periods.

In order to tackle this issue, a modified first-order low-pass filter for NN's outputs is proposed as follows:

$$Y_f = Y_f + \frac{Y - Y_f}{|m(n) - m(n-1)| \times fc + 1} \tag{12}$$

where Y denotes NN's outputs, Y_f denotes the final desired rhythmic signals, $m(n)$ is the number of sample points in the nth period, and fc is a constant to control the smoothness of Y_f. By making use of (12), as shown in Fig. 9, when the frequency of the Matsuoka's CPG output is fixed, Y_f is consistent with Y because of $m(n) = m(n-1)$. However, when the frequency of the Matsuoka's CPG output changes at 226 s, Y becomes discontinuous. At this time, $m(n) \neq m(n-1)$ and the filtering function in (12) starts working. Under this condition, the change of Y_f is slower than that of Y during the transition due to the characteristic of the first-order low-pass filter. As a result, by transforming Y into Y_f, the discontinuous part of Y is smoothed.

Remark 2: It is noteworthy that the nonlinear dynamic system proposed in [31] also learns to generate rhythmic signals with desired shapes. Furthermore, [32] and [33] show that the system in [31] is able to learn the frequency, which is similar to the entrainment property of Matsuoka's CPG. However, to achieve the adaptive walking of humanoid robots, there are two limitations in [31]–[33]. First, it takes several periods to learn the frequency, which may not meet the real-time requirement



(a) (b) (c)

Fig. 10. Framework outputs, the Matsuoka's CPG outputs, and the desired rhythmic signals at a specified frequency. The desired rhythmic signals are similar to (a) $y = 0.25\sin(t)$, (b) $y = 0.25(\sin(t) + \sin(2t))$, and (c) $y = 0.25(\sin(t) + \sin(3t))$, respectively.

for adaptive walking of humanoid robots because the desired frequency in this case commonly changes in a short time due to perturbation. Second, the systems are on open-loop state, without considering feedback pathways. Thus, their capability to generate the desired rhythmic signals to suit the changing environments is limited. In contrast, in our framework, due to feature processing, the advantages of temporal properties of Matsuoka's CPG can be maintained. As pointed out in [8], the frequency adaptation of Matsuoka's CPG is fast. Thus, our framework is suitable for adaptive walking of humanoid robots. Moreover, by integrating robot feedback as a part of NN's feature inputs, our framework allows NN learning to generate desired rhythmic signals for the changing environments. In addition, since the output nodes of NN can be added flexibly, it is easy to learn multiple desired rhythmic signals simultaneously in one network, which is beneficial for the trajectory control of high-dimensional systems like humanoid robots.

IV. Experiments on the Proposed Framework

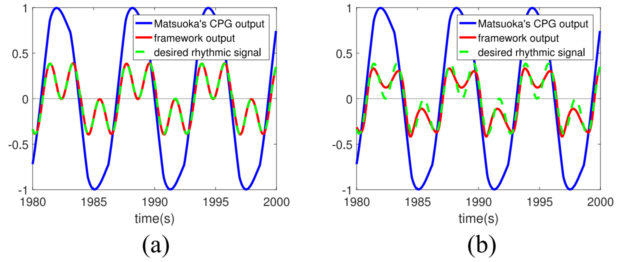
The aim of this section is to empirically investigate whether the proposed framework can generate various desired rhythmic signals. In addition, we reveal several additional advantages of the proposed framework, including the effectiveness of limit cycle normalization, the ability to generate desired rhythmic signals while preserving the entrainment property of Matsuoka's CPG, and the robustness against noise induced by rhythmic feedback. Note that we do not incorporate robot feedback into the proposed framework in this section, but this is discussed in Section V.

A. Generating Desired Rhythmic Signals

In this experiment, the proposed framework learns to generate three desired rhythmic signals simultaneously. The desired rhythmic signals were similar to $y = 0.25\sin(t)$, $y = 0.25(\sin(t) + \sin(2t))$, and $y = 0.25(\sin(t) + \sin(3t))$, respectively. The parameter settings of Matsuoka's CPG were the same as in Table I and $feedback_i$ was set to zero. Then, a frequency of the Matsuoka's CPG outputs was specified. Since the shapes of these desired rhythmic signals were known in advance, the NN learning belongs to supervised learning, and thus SGD [29] was applied for NN weight training. The input layer of NN consisted of two linear neurons; the hidden layer consisted of 25 tanh neurons; and the output layer consisted of three linear neurons. The whole number of sample points for NN learning was 49 500 (1980 s). Fig. 10 shows the framework outputs (the red lines) after learning, which indicates that our framework can generate rhythmic signals approximating the desired rhythmic signals (the green dashed lines) well. However, in these scenarios, Matsuoka's CPG could only generate the rhythmic signals as shown in the blue lines.

TABLE II
Mean SSE and Standard Deviation Provided by Our Framework With and Without Limit Cycle Normalization

τ_2	Our framework with limit cycle normalization Mean SSE±Std Dev	Our framework without limit cycle normalization Mean SSE±Std Dev
3.688575	0.9789±0.9397	45.5537±14.9778
2.76643125	0.6941±0.5959	34.9482±12.0246
1.8442875	0.4500±0.2410	21.6416±6.4499



(a) (b)

Fig. 11. Training results when $\tau_2 = 1.8442875$ in a typical run. (a) Our framework with limit cycle normalization. (b) Our framework without limit cycle normalization.

B. Effectiveness of Limit Cycle Normalization

We compared the precision of the rhythmic signals generated by our framework with and without limit cycle normalization after NN learning. The parameter settings of Matsuoka's CPG were also the same as in Table I except for τ_2, which was tuned for different signal frequencies and different limit cycles (Fig. S-1 in the supplementary file). The desired rhythmic signal was similar to $y = 0.25(\sin(t) + 0.25\sin(3t))$. Like the experiment in Section IV-A, SGD was used as the training method for NN's weights. The input layer of NN consisted of two linear neurons, the hidden layer consisted of 25 tanh neurons, and the output layer consisted of one linear neuron. The whole number of sample points for NN training was 49 500 (1980 s), and 500 points (20 s) were used to compute the sum of squared errors (SSE) for precision comparison. SSE is formulated as (13), where $\hat{y}_i$ and y_i represent the desired signal value and the framework output of the ith sample point,

TABLE III
RHYTHMIC FEEDBACK WITH DIFFERENT FREQUENCIES

Time(s)	$feedback_1$	$feedback_2$
$t < 55$	0	0
$55 \le t < 110$	0.3sin(1.25t)	-0.3sin(1.25t)
$110 \le t < 165$	0.3sin(0.83t)	-0.3sin(0.83t)
$165 \le t < 250$	0.3sin(1.00t)	-0.3sin(1.00t)

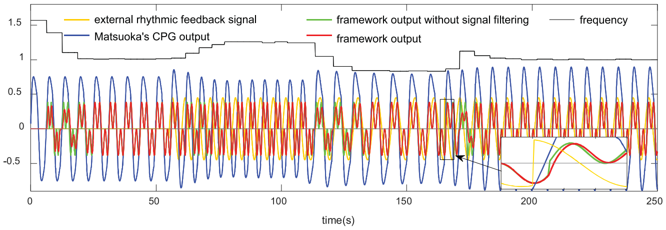


Fig. 12. Entrainment property of the framework output.

respectively

$$\text{SSE} = \sum_{i=1}^{500} (\hat{y}_i - y_i)^2. \tag{13}$$

The average SSE and standard deviation over 50 runs are shown in Table II. It is clear from Table II that our framework with limit cycle normalization achieves higher precision than our framework without limit cycle normalization, which indicates that the uniform distribution of sample points on the limit cycle is definitely beneficial for the performance of NN learning. In addition, the experimental results in Table II also show that our framework with limit cycle normalization can generate desired rhythmic signals with similar precision at different frequencies, regardless of the shapes of limit cycles generated by Matsuoka's CPG (Fig. S-1 in the supplementary file). In Fig. 11, we also presented the experimental results of NN learning when $\tau_2 = 1.8442875$ for our framework with and without limit cycle normalization.

C. Entrainment Property

One may be interested in whether our framework can also preserve the entrainment property of Matsuoka's CPG. As pointed out previously, by introducing the rhythmic feedback into $feedback_i$ in (1), the outputs of Matsuoka's CPG will entrain with the rhythmic feedback. In the proposed framework, we have added three parts into the original Matsuoka's CPG: 1) feature processing; 2) NN learning; and 3) signal filtering. If we can verify that the outputs of our framework can still entrain with rhythmic feedback, then we can argue that our framework has the capability to preserve the entrainment property of Matsuoka's CPG. To this end, we introduced rhythmic feedback with different frequencies into $feedback_i$ after NN learning. In our experiment, the desired rhythmic signal was similar to $y = 0.25\sin(t) + 0.25\sin(3t)$ just as in the experiment in Section IV-B. The parameter settings were the same as in Table I. The settings of rhythmic feedback signals are shown in Table III, and the experimental results are presented in Fig. 12.

TABLE IV
RHYTHMIC FEEDBACK WITH NOISE

Time(s)	$feedback_1$	$feedback_2$
$t < 130$	0.3sin(0.822t)	-0.3sin(0.822t)
$t > 130$	0.3sin(0.822t)+0.3rand(-1,1)	-0.3sin(0.822t)-0.3rand(-1,1)

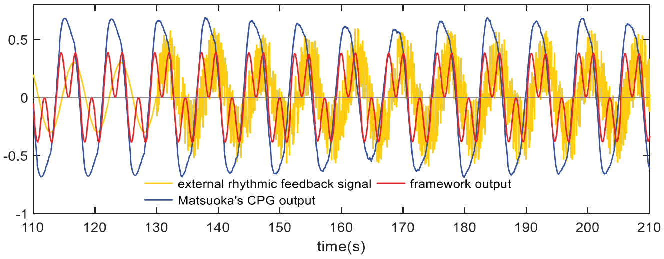


Fig. 13. Robustness of the framework output against the noise from rhythmic feedback.

It can be observed from Fig. 12 that after introducing rhythmic feedback (the yellow line), the entrainment between the Matsuoka's CPG output (the blue line) and the rhythmic feedback appears. Meanwhile, the frequency and phase of the framework output (the red line) track those of the Matsuoka's CPG output well, which indicates that the framework output entrains with the rhythmic feedback. Moreover, unlike the Matsuoka's CPG output, the framework output can still maintain the desired shape when the frequency (the black line) changes. The shape of the framework output slightly changes only in the frequency changing phase because of the signal filtering. For comparison, the framework output without the signal filtering is shown in the green line, which is always consistent with the red line with the exception in the frequency changing phase. Obviously, the red line is smoother than the green line in the frequency changing phase, which also validates the effectiveness of the signal filtering.

D. Robustness Against Noise

When the outputs of Matsuoka's CPG entrain with the rhythmic feedback, the shapes of the outputs might be affected by rhythmic feedback noise, which is harmful to robot control. To test the robustness of our framework against this kind of noise, after NN learned to generate the same desired rhythmic signal as the experiment in Section IV-B, we introduced the rhythmic feedback with noise into Matsuoka's CPG. As shown in Table IV, rand$(-1, 1)$ is the noise, which is a uniformly distributed random number between -1 and 1. The parameter settings of Matsuoka's CPG were the same as in Table I. The experimental results are shown in Fig. 13. From Fig. 13, our framework is robust to the noise since it can maintain the desired shape, while the shape of the Matsuoka's CPG output is affected by the noise.

V. CONTROL SYSTEM

Adaptive walking of humanoid robots is a challenging task in the robotics field. The control trajectories for humanoid robot walking must be adjustable and precise enough to ensure the adaptability and stability in various environments. However, the control system for humanoid robot walking

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

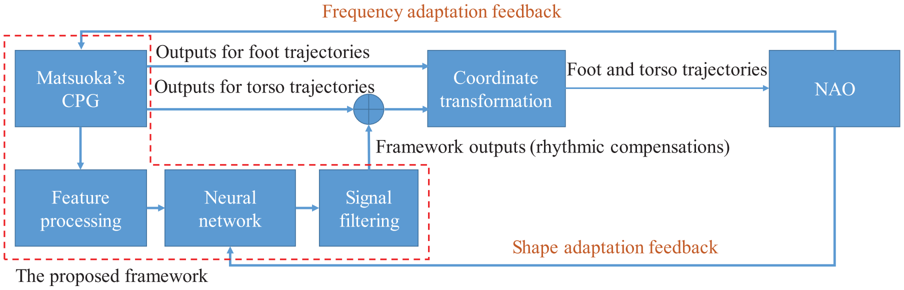

Frequency adaptation feedback
Matsuoka's CPG
Outputs for foot trajectories
Outputs for torso trajectories
Coordinate transformation
Foot and torso trajectories
NAO
Framework outputs (rhythmic compensations)
Feature processing
Neural network
Signal filtering
Shape adaptation feedback
The proposed framework

Fig. 14. Control system for adaptive omnidirectional walking of NAO.

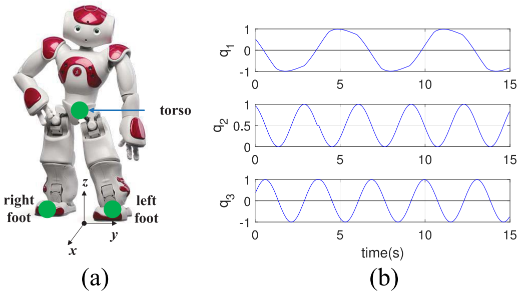


Fig. 15. NAO and the outputs of Matsuoka's CPG. (a) NAO's end effectors and the world frame. (b) Outputs of Matsuoka's CPG.

based on Matsuoka's CPG is difficult to generate various desired control trajectories with high precision. Hence, in this section, we embedded the proposed framework into a control system based on Matsuoka's CPG, with the aim of overcoming this challenge and generating various desired control trajectories for humanoid robot NAO, which was walking on different inclined surfaces.

A. Overview of the Control System

Fig. 14 shows the control system. The control strategy in this system is similar to the one used in [14], which considers the Matsuoka's CPG outputs as the trajectories of NAO's end effectors. Matsuoka's CPG in this system consisted of two neurons. We only used three Matsuoka's CPG outputs to control the trajectories of NAO's end effectors, i.e., right foot, left foot, and torso. These trajectories are based on the world frame, as shown in Fig. 15(a). The three Matsuoka's outputs (i.e., q_1, q_2, and q_3) are given in (14)–(16), where tp_i is calculated according to (9)

$$q_1 = y_1 - y_2 \tag{14}$$

$$q_2 = -0.5\sin\left(\frac{4\pi tp_i}{T}\right) + 0.5 \tag{15}$$

$$q_3 = -\cos\left(\frac{4\pi tp_i}{T}\right). \tag{16}$$

Fig. 15(b) depicts the values of q_1, q_2, and q_3.

Instead of only walking straight as in [14], our control system allowed NAO to walk omnidirectionally. To achieve this, a coordinate transformation from the Matsuoka's CPG outputs to NAO's foot and torso trajectories was designed, which is given in Appendix 1 of the supplementary file.

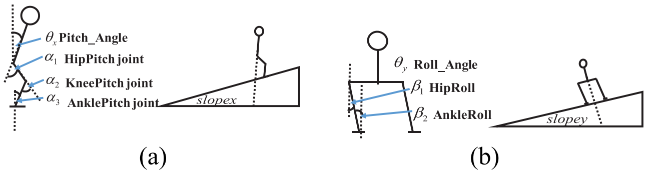


Fig. 16. (a) Real slope in the sagittal plane. (b) Real slope in the lateral plane.

The framework presented in Section III was embedded into this control system, enabling it to generate the desired trajectories for adaptive walking of NAO. The framework outputs acted as rhythmic compensations for the Matsuoka's CPG outputs. For simplicity, we only considered the rhythmic compensations for torso trajectories, which are essential for the stability and adaptability of NAO when walking on different surfaces. Thus, the framework output 3-D rhythmic compensations corresponding to the 3-D torso trajectories.

Additionally, two kinds of robot feedback were introduced to further module the Matsuoka's CPG outputs and the rhythmic compensations. These were the shape adaptation feedback for NN and the frequency adaptation feedback for Matsuoka's CPG. For the shape adaptation feedback, two types of fusion sensory information were used. The first type was the slope in the sagittal plane (*slopex*), which indicates the real tilt angle of the ground in the sagittal plane with respect to the flat ground, as shown in Fig. 16(a) and formulated as

$$slopex = \alpha_1 + \alpha_3 - \alpha_2 - \theta_x \tag{17}$$

where α_1, α_2, and α_3 are three joint angles and θ_x is the tilt angle of NAO's upper body in the sagittal plane. In addition, the second type was the slope in the lateral plane (*slopey*), which indicates the real tilt angle of the ground in the lateral plane with respect to the flat ground, as shown in Fig. 16(b) and formulated as

$$slopey = \beta_1 - \beta_2 + \theta_y \tag{18}$$

where β_1 and β_2 are two joint angles and θ_y is the tilt angle of NAO's upper body in the lateral plane. These two types of fusion sensory information were able to make the framework generate rhythmic compensations for NAO to suit the inclined surfaces. On the other hand, as far as the frequency adaptation feedback is concerned, the rhythmic feedback ($feedback_1$ and $feedback_2$) from NAO's mechanical system were introduced

$$feedback_1 = FsrR - FsrL \tag{19}$$

$$feedback_2 = -feedback_1 \tag{20}$$

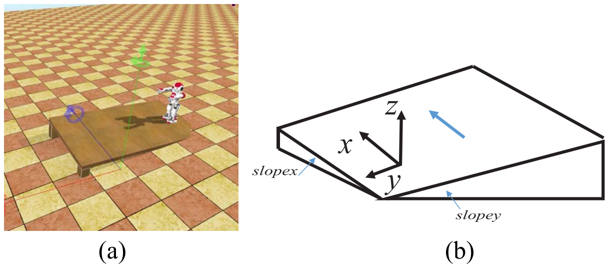


Fig. 17. Fixed inclined surface for NAO's walking.

where *FsrR* and *FsrL* indicate the pressure signals from NAO's right and left feet, respectively. By integrating this kind of feedback, the outputs of the control system could entrain with the dynamics of NAO's mechanical system, thus enhancing the robustness of NAO's walking.

B. Desired Rhythmic Compensations

In this control system, the desired trajectory of NAO's COP, which ensures that NAO walks stably on different surfaces, is considered as the learning objective to evaluate the rhythmic compensations generated by our framework. The difference between the desired and the actual COP trajectories is formulated as

$$D_{COP} = \sum_{t=0}^{T_{ls}} \left| x_d(t) - COP_x(t) \right| + \left| y_d(t) - COP_y(t) \right| \quad (21)$$

where $(x_d(t), y_d(t))$ is the desired COP trajectory, and $(COP_x(t), COP_y(t))$ is the actual COP trajectory in NAO's life span T_{ls}. In (22), *penalty* indicates the punishment for NAO's falling down

$$penalty = \left(1 - \frac{t_f}{T_{ls}}\right) \quad (22)$$

where t_f is the moment when NAO falls down.

The final objective function (i.e., *fitness*) for evaluating the rhythmic compensations generated by our framework can be expressed as

$$\text{maximize}\ \ fitness = \frac{\omega_1}{D_{COP}} - \omega_2 penalty \quad (23)$$

where ω_1 and ω_2 are two coefficients. In (23), the more accurate the actual COP trajectory tracking the desired one and the longer NAO walking stably without falling down, the higher the value of *fitness*. By maximizing *fitness*, we can obtain the desired rhythmic compensations.

Due to the fact that the desired rhythmic compensations cannot be known *a priori*, an evolutionary algorithm is a preferred method for NN weight training. During the past two decades, DE has become one of the most popular evolutionary algorithm paradigms and has been successfully applied to solve numerous optimization problems in different fields [21], [34], [35]. DE has strong global search ability. In this paper, we applied DE to train NN's weights to find the desired rhythmic compensations.

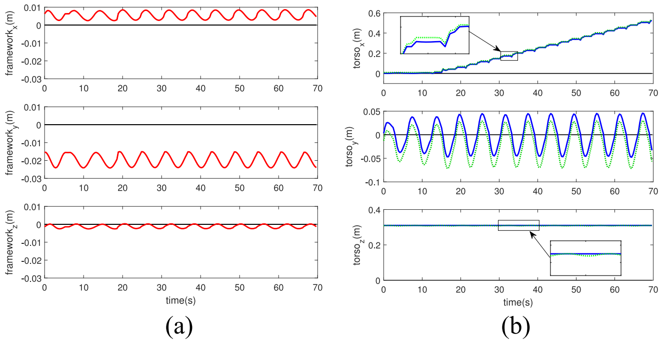


Fig. 18. (a) Rhythmic compensations generated by our framework. (b) Torso trajectories after DE optimization, in which the green dashed lines indicate the torso trajectories with rhythmic compensations, and the blue lines indicate the torso trajectories without rhythmic compensations.

VI. Experiments on Control System

This section aims at verifying the effectiveness of the control system for adaptive walking of humanoid robots. Specifically, we investigated whether the proposed framework can generate the desired rhythmic compensations to make NAO adaptively walk not only on a fixed inclined surface but also on a changing inclined surface.

A. Adaptive Walking of NAO on Fixed Inclined Surface

1) Experiment in Simulation: First, we considered a fixed inclined surface. A slope environment in Webots [36] was set for NAO to walk on, in which the inclined angle was set to 8.592°; thus, the inclined angles in both the sagittal and lateral planes were 6.077° (i.e., $slopex = slopey = 6.077°$), as shown in Fig. 17. The desired COP trajectory $(x_d(t), y_d(t))$ is presented in Appendix 2 of the supplementary file. Through the experimental studies, the parameter settings of (21)–(23) were the following: $T_{ls} = 70$ s, $\omega_1 = 1.4$, and $\omega_2 = 2$. The population size of DE was set to 30, and the total generation number was set to 45. The parameter settings of Matsuoka's CPG were the same as in Table I; thus, the period of the control trajectories generated by the control system was about 6 s. The parameter settings of the coordinate transformation in Appendix 1 of the supplementary file were the following: $stride_x = 0.02$, $stride_y = 0.1$, and $\Delta\theta = 0$. Therefore, the walking speed of NAO was about 0.0133 m/s. By adjusting $stride_x$ to a bigger stride, a greater speed can be obtained. The outputs of the coordinate transformation for foot and torso trajectories under these parameter settings are shown in Fig. S-2 of the supplementary file.

After DE optimization, Fig. 18 shows the rhythmic compensations generated by our framework [the red lines in Fig. 18(a)] and torso trajectories generated by the control system [the green dashed lines in Fig. 18(b)] with respect to the best value of *fitness*. Under this condition, *slopex* and *slopey*, which are the slopes calculated by NAO in x and y directions (i.e., the sagittal and lateral planes), are shown in Fig. S-3 of the supplementary file. They are used as the shape adaptation feedback in Fig. 14.

It can be seen that the rhythmic compensations generated by our framework coincide with the characteristics of the slope

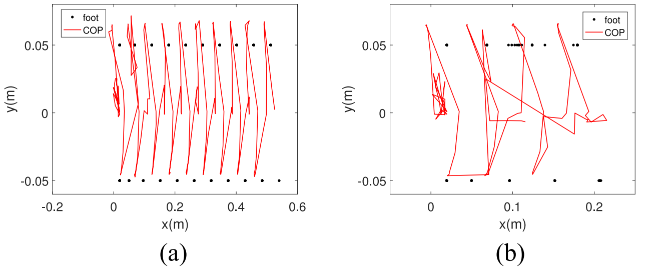


Fig. 19. (a) Actual COP trajectory, in which the black points are the positions of NAO's feet. (b) COP trajectory when the control system excludes the rhythmic compensation generated by our framework.

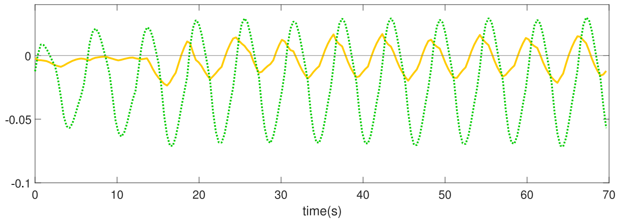


Fig. 20. Frequency synchronization between the robot rhythmic feedback (the yellow line) and the torso trajectory in y direction (the green dashed line).

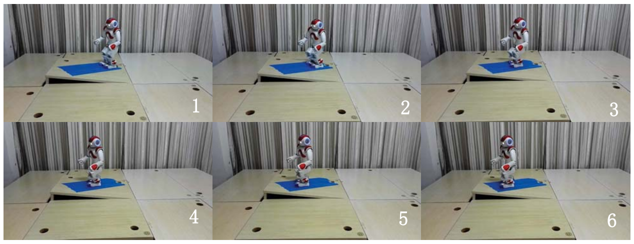


Fig. 21. Snapshots of NAO's walking on a fixed inclined surface in the real world.

environment shown in Fig. 17. That is, the rhythmic compensation generated by our framework in x direction is positive and in y direction is negative. Thus, the corresponding torso trajectory in x direction is higher than the torso trajectory without rhythmic compensation [shown in the top subgraph of Fig. 18(b)], and the corresponding torso trajectory in y direction is lower than the torso trajectory without rhythmic compensation [shown in the middle subgraph of Fig. 18(b)]. In addition, due to the small value of rhythmic compensation in z direction, the torso trajectories with and without rhythmic compensation in z direction are similar [shown in the bottom subgraph of Fig. 18(b)]. The above phenomenon is consistent with the general way of compensation for adaptive slope walking of humanoid robots [14], [15], [37]–[39]. From the above discussion, we can conclude that, on the fixed inclined surface, the stability of NAO's walking depends mainly on the rhythmic compensations in x and y directions, but relies less on the rhythmic compensation in z direction.

The COP trajectory corresponding to the control trajectories in Fig. 18 is shown in Fig. 19(a), which indicates that NAO can stably walk on the fixed inclined surface. As a comparative observation, Fig. 19(b) shows the COP trajectory when the control system excludes the rhythmic compensations generated by our framework, which suggests that NAO cannot walk adaptively on this fixed inclined surface and instead falls down.

TABLE V
MAXIMAL *fitness* OBTAINED BY THREE METHODS ON DIFFERENT FIXED INCLINED SURFACES

Slope angle(degree)	Our method	The method in [14] improved in this paper	The original method in [14]
0	0.400884	0.362114	0.34556
2	0.346707	0.295156	0.281
4	0.291233	0.255359	0.252342
6	0.271645	0.227979	–
8.592	0.232803	0.208493	–
10	0.200076	–	–
12	0.193161	–	–

In Section IV-C, we have verified that our framework exhibits the entrainment property. A question which arises naturally is whether the control system still has this property after integrating with our framework. To this end, we tested the frequency synchronization between the control system output and NAO's rhythmic feedback. The experimental results are shown in Fig. 20, in which the yellow line denotes the robot rhythmic feedback in (19) and (20) and the green dashed line is the torso trajectory in y direction ($torso_y$) after introducing the robot rhythmic feedback. It can be seen that after about 15 s, the robot rhythmic feedback is activated, and the frequency of the torso trajectory in y direction is automatically adjusted to be synchronized with the frequency of the robot rhythmic feedback. Therefore, the control system maintains the entrainment property.

2) Experiment in the Real World: Based on the simulation results, we made use of the control system to control the real NAO. The angle of the real inclined surface was the same as in the simulation. Snapshots of the experiment are shown in Fig. 21, which indicate that NAO could also realize a stable and adaptive walk on a fixed inclined surface in the real world.

3) Further Comparison With Other Methods: To further demonstrate the advantage of our control system, comparison studies were conducted. Among CPG-based control methods, the method in [14] has been a competitive method for adaptive walking of humanoid robots in recent years. Due to its similar control strategy, the method in [14] was chosen as the compared method. Note, however, that it only adds the compensation derived from the tilt angle of NAO's upper body θ_x for torso trajectory in x direction, thus achieving adaptive walking of NAO on the sagittal plane. To ensure a fair comparison, we made some improvements to the method in [14] to enable NAO to walk on both the sagittal and lateral planes. To achieve this, the compensations for torso trajectories in x and y directions were given in (24) and (25), and the compensation for torso trajectory in z direction was still set to zero

$$comp_x = k_1 slopex \tag{24}$$

$$comp_y = k_2 slopey \tag{25}$$

where $comp_x$ and $comp_y$ represent the compensations for $torso_x$ and $torso_y$, respectively, and k_1 and k_2 are two adjustable parameters. Note that k_1 and k_2 have an influence on the adaptability of humanoid robots and were also optimized

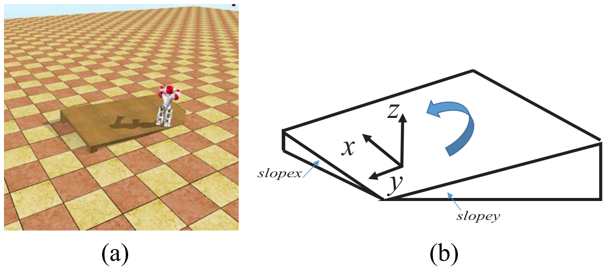


Fig. 22. Changing inclined surface for NAO's circle walking.

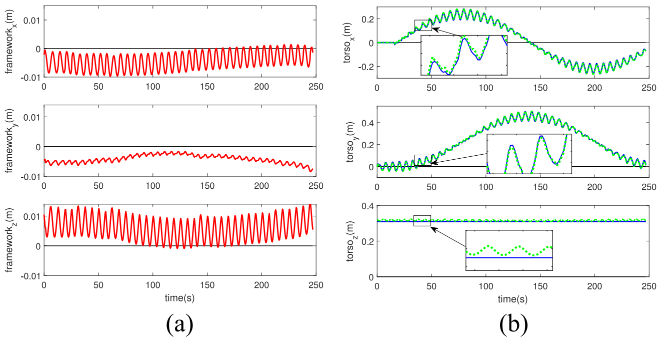


Fig. 23. (a) Rhythmic compensations generated by our framework. (b) Torso trajectories after DE optimization, in which the green dashed lines indicate the torso trajectories with rhythmic compensations, and the blue lines indicate the torso trajectories without rhythmic compensations.

by DE in this experiment to maximize *fitness* in (23). For comparison, we also reported the performance of the original method in [14].

In our experiment, different angles of the fixed slope were tested. The experimental results are presented in Table V, which shows the maximal *fitness* obtained by different methods over 20 independent runs. In Table V, "—" means that NAO falls down. As shown in Table V, our method consistently outperforms the two competitors. By implementing our method, NAO could stably and adaptively walk on various fixed inclined surfaces. Moreover, the experimental results in Table V suggest that the performance of the method in [14] could be improved by introducing the compensations for torso trajectories in x and y directions.

B. Adaptive Walking of NAO on Changing Inclined Surface

1) Experiment in Simulation: To verify that the proposed framework can generate desired rhythmic compensations for NAO's adaptive walking on a changing inclined surface, we set a slope environment with a fixed inclined angle of 4.58° as shown in Fig. 22. In the mean time, we made NAO walk the circle on this inclined surface, and thus the slope was always changing for NAO. By doing this, the slope environment was a changing inclined surface for NAO. The desired COP trajectory $(x_d(t), y_d(t))$ is presented in Appendix 2 of the supplementary file. Through the experimental studies, the parameters in (21)–(23) were set as $T_{ls} = 250$ s, $\omega_1 = 1.4$, and $\omega_2 = 2$. The population size for DE was equal to 30, and the total generation number was set to 60. The parameter settings of Matsuoka's CPG were the same as in Table I; thus, the period of the control trajectories generated by the control system was about 6 s. The parameters settings of the coordinate transformation in Appendix 1 of the supplementary file were the following: $stride_x = 0.02$, $stride_y = 0.1$, and $\Delta\theta = \pi/36$. Therefore, the walking speed of NAO was about 0.0133 m/s, and the radius of the circle of NAO's walking was about 0.2 m. Again, by adjusting $stride_x$ to a bigger stride, a greater speed can be reached. Moreover, by adjusting $\Delta\theta$, different radii of the circle of NAO's walking can be achieved. The outputs of the coordinate transformation for foot and torso trajectories under these parameter settings are shown in Fig. S-4 of the supplementary file.

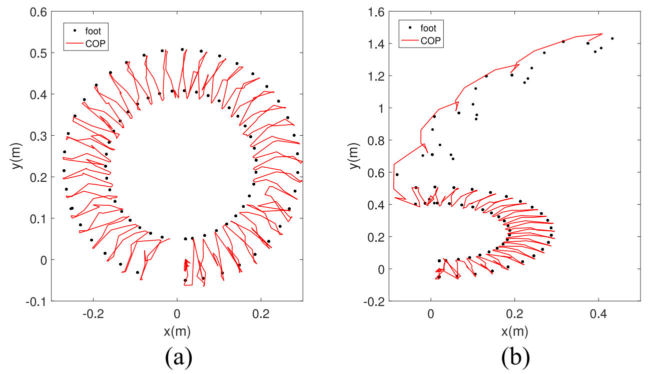


Fig. 24. (a) Actual COP trajectory when NAO walks the circle on a changing inclined surface. The COP trajectory is always located in the support region (between the black points), and the black points are the positions of NAO's feet. (b) When the control system excludes the rhythmic compensation generated by our framework, the COP trajectory deviates from the support region, and finally NAO falls down.

After DE optimization, the rhythmic compensations generated by our framework and the torso trajectories generated by the control system corresponding to the best value of *fitness* are shown in the red and green dashed lines in Fig. 23, respectively. Under this condition, *slopex* and *slopey* are shown in Fig. S-5 of the supplementary file. It is noteworthy that *slopex* and *slopey* are used as the shape adaptation feedback in Fig. 14.

Unlike the experimental results in Fig. 18(a) and Fig. S-3 of the supplementary file, in which rhythmic compensations in x and y directions depend on *slopex* and *slopey*, respectively, the experimental results of Fig. 23(a) and Fig. S-5 in the supplementary file show that rhythmic compensations in x and y directions are dependent on both *slopex* and *slopey*. Actually, when NAO adaptively walks the circle on the changing inclined surface, both *slopex* and *slopey* have an influence on the rhythmic compensations in x, y, and z directions.

In addition, according to the general way of compensation for adaptive slope walking of humanoid robots [14], [15], [39], when humanoid robots adaptively walk on a fixed inclined surface, the rhythmic compensations for torso trajectories in x and y directions are large, and the rhythmic compensation for torso trajectory in z direction is small. The above phenomenon has already been demonstrated in Section VI-A. However, interestingly, from Fig. 23, we observe the opposite phenomenon on this changing inclined surface, that is, the rhythmic compensations for torso trajectories in x and y

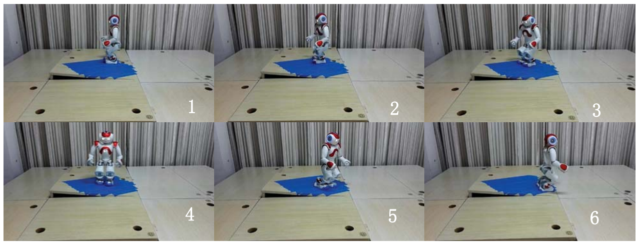


Fig. 25. Snapshots of NAO's walking on a changing inclined surface in the real world.

TABLE VI
MAXIMAL *fitness* OBTAINED BY THREE METHODS ON DIFFERENT CHANGING INCLINED SURFACES

Slope angle(degree)	Our method	The method in [14] improved in this paper	The original method in [14]
0	0.158031	0.153600	0.149815
2	0.154490	0.146005	0.143387
4	0.1497225	–	–
6	0.1293387	–	–
8	0.058277	–	–

directions are small, and the rhythmic compensation for torso trajectory in z direction is large.

The actual COP trajectory $(COP_x(t), COP_y(t))$ corresponding to the optimized torso trajectories is shown in Fig. 24(a), which confirms that NAO is able to walk stably on the changing inclined surface. As a comparative observation, Fig. 24(b) shows the COP trajectory when the control system excludes the rhythmic compensation generated by our framework, which indicates that NAO cannot walk the circle on the changing inclined surface and finally falls down.

2) Experiment in the Real World: We also used the control system based on the simulation experiments to control the real NAO. The angle of the real inclined surface was the same as in the simulation. Fig. 25 shows the snapshots of the experiment, which suggest that NAO can also walk on a changing inclined surface in the real world stably and adaptively.

3) Further Comparison With Other Methods: We also conducted comparison experiments on several changing inclined surfaces with different slope angles. Similar to Section VI-A, the original method in [14] and its improved version in this paper were chosen as the compared methods. Table VI summarizes the maximal *fitness* provided by different methods over 20 independent runs. In Table VI, "—" means that NAO falls down. From Table VI, our method exhibits the best performance in all the cases and makes NAO stably and adaptively walk on more changing inclined surfaces. Again, the experimental results in Table VI verify that the compensations for torso trajectories in x and y directions can further improve the performance of the method in [14].

VII. CONCLUSION

In this paper, a framework that can learn to generate desired rhythmic signals for Matsuoka's CPG has been presented. The framework can not only generate rhythmic signals with desired shapes but also preserve the frequency and phase properties of Matsuoka's CPG. By introducing the robot feedback, the framework is capable of generating desired rhythmic signals corresponding to different environments. Furthermore, by embedding this framework into a control system, adaptive walking was successfully achieved by humanoid robot NAO in terms of both fixed and changing inclined surfaces.

Although the main aim of this paper is to overcome the drawback that Matsuoka's CPG cannot generate rhythmic signals with desired shapes, this issue also exists in other kinds of CPG. Actually, the proposed framework is also feasible for other kinds of CPG (e.g., Hopf [40] and VDP [41]). It is because the irregular limit cycles of Hopf and VDP can also be transformed into the normalized limit cycle via feature processing, which can be subsequently used as the feature input of NN for learning to generate desired rhythmic signals. We believe that this paper will enhance the reliability of CPG's application in robot movement control. As our future work, more powerful control systems based on the proposed framework will be designed for adaptive walking of humanoid robots on various challenging environments.

The videos of the experiments in the real world can be downloaded from: http://www.escience.cn/people/yongwang1.

REFERENCES

[1] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Netw*, vol. 21, no. 4, pp. 642–653, 2008.

[2] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on CPG-inspired control models and system implementation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 441–456, Mar. 2014.

[3] C.-F. Juang and Y.-T. Yeh, "Multiobjective evolution of biped robot gaits using advanced continuous ant-colony optimized recurrent neural networks," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1910–1922, Jun. 2018.

[4] K. Matsuoka, "Sustained oscillations generated by mutually inhibiting neurons with adaptation," *Biol. Cybern.*, vol. 52, no. 6, pp. 367–376, 1985.

[5] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biol. Cybern.*, vol. 56, nos. 5–6, pp. 345–353, 1987.

[6] K. Matsuoka, "Analysis of a neural oscillator," *Biol. Cybern.*, vol. 104, nos. 4–5, pp. 297–304, 2011.

[7] M. M. Williamson, "Neural control of rhythmic arm movements," *Neural Netw.*, vol. 11, nos. 7–8, pp. 1379–1394, 1998.

[8] G. Endo, J. Nakanishi, J. Morimoto, and G. Cheng, "Experimental studies of a neural oscillator for biped locomotion with QRIO," in *Proc. Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 596–602.

[9] G. Taga, Y. Yamaguchi, and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biol. Cybern.*, vol. 65, no. 3, pp. 147–159, 1991.

[10] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng, "Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 213–228, Feb. 2008.

[11] Y. Fukuoka, H. Kimura, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *Int. J. Robot. Res.*, vol. 22, nos. 3–4, pp. 187–202, 2003.

[12] X. Zhang and H. Zheng, "Walking up and down hill with a biologically-inspired postural reflex in a quadrupedal robot," *Auton. Robots*, vol. 25, nos. 1–2, pp. 15–24, 2008.

[13] C. Liu, Q. Chen, and D. Wang, "CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 867–880, Jun. 2011.

[14] C. Liu, D. Wang, and Q. Chen, "Central pattern generator inspired control for adaptive walking of biped robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 5, pp. 1206–1215, Sep. 2013.

[15] C. Liu, D. Wang, D. Goodman, and Q. Chen, "Adaptive walking control of biped robots using online trajectory generation method based on neural oscillators," *J. Bionic Eng.*, vol. 13, no. 4, pp. 572–584, 2016.

[16] C.-S. Park, Y.-D. Hong, and J.-H. Kim, "Full-body joint trajectory generation using an evolutionary central pattern generator for stable bipedal walking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 160–165.

[17] C.-S. Park, Y.-D. Hong, and J.-H. Kim, "Evolutionary-optimized central pattern generator for stable modifiable bipedal walking," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1374–1383, Aug. 2014.

[18] Y.-D. Hong, C.-S. Park, and J.-H. Kim, "Stable bipedal walking with a vertical center-of-mass motion by an evolutionary optimized central pattern generator," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2346–2355, May 2014.

[19] A. A. Saputra, J. Botzheim, I. A. Sulistijono, and N. Kubota, "Biologically inspired control system for 3-D locomotion of a humanoid biped robot," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 898–911, Jul. 2016.

[20] Y.-D. Hong and B. Lee, "Evolutionary optimization for optimal hopping of humanoid robots," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1279–1283, Feb. 2017.

[21] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[22] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[23] J. Ma and J. Tang, "A review for dynamics of collective behaviors of network of neurons," *Sci. China Technol. Sci.*, vol. 58, no. 12, pp. 2038–2045, 2015.

[24] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biol. Cybern.*, vol. 95, no. 1, pp. 1–19, 2006.

[25] J.-H. He, "Limit cycle and bifurcation of nonlinear problems," *Chaos Solitons Fractals*, vol. 26, no. 3, pp. 827–833, 2005.

[26] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989.

[27] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[28] Y. Anzai, *Pattern Recognition and Machine Learning*, Elsevier, 2012.

[29] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—Online stochastic gradient for tensor decomposition," in *Proc. 28th Conf. Learn. Theory*, 2015, pp. 797–842.

[30] L. Wang, Y. Zeng, and T. Chen, "Back propagation neural network with adaptive differential evolution algorithm for time series forecasting," *Expert Syst. Appl.*, vol. 42, no. 2, pp. 855–863, 2015.

[31] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning rhythmic movements by demonstration using nonlinear oscillators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, pp. 958–963.

[32] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarcic, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Auton. Robots*, vol. 27, no. 1, pp. 3–23, 2009.

[33] T. Petrič, A. Gams, and A. J. Ijspeert, "On-line frequency adaptation and movement imitation for rhythmic robotic tasks," *Int. J. Robot. Res.*, vol. 30, no. 14, pp. 1775–1788, 2011.

[34] Y. Wang, B. Xu, G. Sun, and S. Yang, "A two-phase differential evolution for uniform designs in constrained experimental domains," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 665–680, Oct. 2017.

[35] Y.-Z. Zhou, W.-C. Yi, L. Gao, and X.-Y. Li, "Adaptive differential evolution with sorting crossover rate for continuous optimization problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2742–2753, Sep. 2017.

[36] O. Michel, "Webots: Professional mobile robot simulation," *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, pp. 39–42, 2004.

[37] J. Cristiano, D. Puig, and M. A. Garcia, "Efficient locomotion control of biped robots on unknown sloped surfaces with central pattern generators," *Electron. Lett.*, vol. 51, no. 3, pp. 220–222, May 2015.

[38] K.-T. Song and C.-H. Hsieh, "CPG-based control design for bipedal walking on unknown slope surfaces," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 5109–5114.

[39] Z. Yu *et al.*, "Gait planning of omnidirectional walk on inclined ground for biped robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 888–897, Jul. 2016.

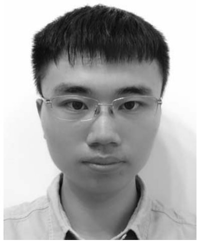
[40] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic Hebbian learning in adaptive frequency oscillators," *Physica D Nonlin. Phenom.*, vol. 216, no. 2, pp. 269–281, 2006.

[41] H. Yu *et al.*, "Gait generation with smooth transition using CPG-based locomotion control for hexapod walking robot," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5488–5500, Sep. 2016.

Yong Wang (M'08–SM'17) received the Ph.D. degree in control science and engineering from Central South University (CSU), Changsha, China, in 2011.

He is currently a Professor with the School of Information Science and Engineering, CSU. His current research interests include the theory, algorithm design, and interdisciplinary applications of computational intelligence.

Dr. Wang was a recipient of the Hong Kong Scholar by the Mainland-Hong Kong Joint Post-Doctoral Fellows Program, China, in 2013, the Excellent Doctoral Dissertation by Hunan Province, China, in 2013, the New Century Excellent Talents in University by the Ministry of Education, China, in 2013, the 2015 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars, in 2016, the EU Horizon 2020 Marie Sklodowska-Curie Fellowship, in 2016, and a Highly Cited Researcher Award in computer science by Clarivate Analytics, in 2017 and 2018. He is currently serving as an Associate Editor for the *Swarm and Evolutionary Computation*.



Xihui Xue received the B.S. degree in automation and the M.S. degree in control science and engineering from Central South University, Changsha, China, in 2015 and 2018, respectively.

He is researching on LIDAR-based perception algorithm with SF Technology, Shenzhen, China. His current research interests include humanoid robots, reinforcement learning, and 3-D data processing.

Baifan Chen (M'12) received the Ph.D. degree in computer science from Central South University (CSU), Changsha, China, in 2009.

She is currently an Associate Professor with the School of Information Science and Engineering, CSU. She was a Visiting Scholar with the NetBot Laboratory, Texas A&M University, College Station, TX, USA, in 2015. Her current research interests include mobile robot, machine learning, and SLAM.