

# Prescription of rhythmic patterns for legged locomotion

Zhijun Yang<sup>1</sup> · Daqiang Zhang<sup>2</sup> · Marlon V. Rocha<sup>3</sup> · Priscila M. V. Lima<sup>4</sup> · Mehmet Karamanoglu<sup>1</sup> · Felipe M. G. França<sup>3</sup>

Received: 23 July 2015 / Accepted: 16 February 2016 / Published online: 16 March 2016  
© The Natural Computing Applications Forum 2016

**Abstract** As the engine behind many life phenomena, motor information generated by the central nervous system plays a critical role in the activities of all animals. In this work, a novel, macroscopic and model-independent approach is presented for creating different patterns of coupled neural oscillations observed in biological central pattern generators (CPG) during the control of legged locomotion. Based on a simple distributed state machine, which consists of two nodes sharing pre-defined number of resources, the concept of oscillatory building blocks (OBBs) is summarised for the production of elaborated rhythmic patterns. Various types of OBBs can be designed to construct a motion joint of one degree of freedom with adjustable oscillatory frequencies and duty cycles. An OBB network can thus be potentially built to generate a full range of locomotion patterns of a legged animal with controlled transitions between different rhythmic patterns. It is shown that gait pattern transition can be achieved by simply changing a single parameter of an OBB module. Essentially, this simple mechanism allows for the consolidation of a methodology for the construction of artificial CPG architectures behaving as an asymmetric Hopfield neural network. Moreover, the proposed CPG model introduced here is amenable to analogue and/or digital circuit integration.

**Keywords** Central pattern generator · Oscillatory building blocks · Legged locomotion · Parallel processing systems

## 1 Introduction

Animal locomotion is generated and controlled, arguably, by central pattern generators (CPGs), which are networks of neurons in the central nervous system (CNS) capable of producing rhythmic outputs, usually as a result of interactions between CNS and external stimulation via a range of bio-sensors [1–5]. Although current neurophysiological techniques have difficulties in clearly isolating such circuits from the intricate neural connections of animals, the indirect experimental evidence for their existence is strong [6–16]. The CPGs have been studied and modelled in details in terms of their biological significance [17–25], stringent mathematical forms [26–34], and different animal species [35–45]. Furthermore, CPGs mechanisms have already been implemented in mixed-signal circuit chips and applied to build autonomous robots [46–60]. In these studies, the locomotion patterns are supposed to be the outputs of musculoskeletal systems driven by CPGs, whose parameters and functions are adjusted by neuro-modulators or tonic inputs [61]. Although the CNS mechanism underlying CPGs is not entirely clear, artificial models have been widely applied to map the possible functional organisation of the CPGs network onto the motor system responsible for driving locomotion.

A motor system is usually modelled by coupled oscillators, which represent the activity of neurophysiologically simplified motor neurons. Different types of oscillators can be chosen and connected, usually with a topological shape to simulate a specific animal species [63–65]. The

✉ Zhijun Yang  
Z.Yang@mdx.ac.uk

<sup>1</sup> School of Science and Technology, Middlesex University, London NW4 4BT, UK

<sup>2</sup> School of Software, Tongji University, Shanghai, China

<sup>3</sup> Systems Engineering and Computer Science Program, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

<sup>4</sup> Tércio Pacitti Institute, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

connection strength in the oscillator network is usually adaptive based on the external inputs or internal, CNS instructions. The nature of the parallel and distributed processing is a prominent characteristic of the circuit that can be canonically described by a group of ordinary differential equations, which usually reflect an autonomous system.

So far many coupled, nonlinear oscillator models have been suggested on CPG mechanisms for vertebrate and invertebrate animals, for instance, the biped [62, 65–67], quadruped [68–71], and hexapod models [74, 75]. Most of them use one or more units of CPGs for generating and switching among gait patterns. For instance, Schöner and colleagues [68] used a synergetic approach to study the quadrupedal locomotion. Synergy deals with cooperative phenomena. In synergy, the macroscopic behaviour of a complex system can be characterised by a small number of collective variables which in turn govern the qualitative behaviour of the system elements [72]. In [68], a network model, which consists of four coupled oscillators, is analysed. Each oscillator represents a limb of a quadruped model. The phase difference among limbs is used as collective variables to characterise the inter-limb coordinating patterns of this discrete system. Gait transitions are simply modelled as phase transitions, which can also be interpreted as bifurcations in a dynamical system. This approach is significant in the sense that it relates the system parameter changes and stability issues to gait transitions. CPG models with the similar synergetic operations have also been proposed and implemented in circuits [53, 54].

Inspired by the synergetic CPGs models, a novel, structural approach to the modelling of the complex, legged locomotion is presented based on a graph dynamics. In this approach, a concept, namely the oscillatory building blocks (OBBs) [73–75], is updated and its generalised version presented. By selecting and building with pre-configured OBBs modules, different gait patterns can be achieved for producing coordinated gait patterns, constructing locomotion prototypes and facilitating the circuit synthesis in an efficient, uniform, and systematic framework. Built upon our previous study on OBB modules, this work generalises it by introducing a mathematical CPGs framework for both gait pattern generation and transition between different patterns. The transitions can be realised by simply changing a few of system parameters. A similar prototype of the proposed OBB modules has already been implemented in an integrated circuit [57, 58], in which one control signal is used to change the gait patterns.

The work is organised in four sections. The following section presents a detailed model description, including the graph dynamics, the generalised concept of OBB modules, their architecture, and biological significance. Section 3 provides a simple computer simulation of gait pattern

generation and transition as a case study using the proposed OBB modules, which is followed by a conclusion.

## 2 Method

This section presents the graph dynamics, namely Scheduling by Edge Reversal (SER) [76, 77], and its generalisation, namely Scheduling by Multiple Edge Reversal (SMER) [78, 79]. The dynamics of the discrete and analogue OBB modules are then introduced. It is shown that the discrete OBB module can be used to simulate the envelope shape of the relaxation oscillation of two coupled neurons in an intuitive way. A more elaborate method to simulate the oscillatory patterns and their transitions is to use analogue OBB modules and networks, as introduced in this section.

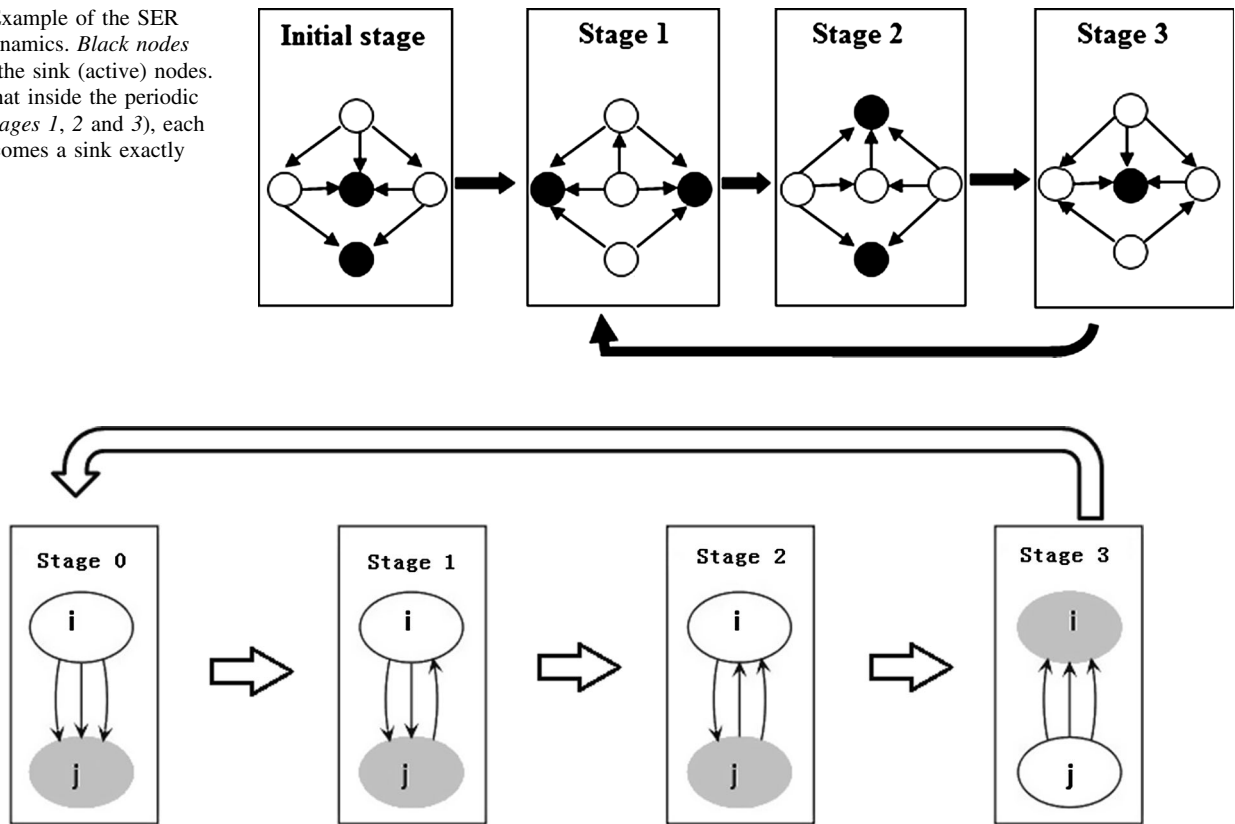
### 2.1 Graph dynamics

The SER graph dynamics considers a neighbourhood-constrained system represented by a set of nodes (or neurons) and by a set of directed edges with one edge for a pair of connected nodes, i.e. a directed graph  $G = (N, E)$  where  $N$  is the set of nodes,  $E$  is the set of edges defining the interconnected topology. In SER, each pair of nodes has at most one edge connecting them.

In order to produce a periodic behaviour, the SER dynamics assume that  $G$  is acyclic, i.e. that node  $i$  cannot be reached by following any directed path starting from itself. The stage transition occurs through the reversal of the direction of all the edges of every node having all of edges directed to it. This node is thus referred to as a sink. After edge reversal every sink node becomes a source node, which has all of its edges directed away from it. The resulting directed graph is also acyclic. The edge reversal operation will continue until a repetition of an acyclic stage is reached. This periodic behaviour can be seen as a dynamic attractor. SER is simple and is fully distributed graph dynamics. A very interesting property of this algorithm is that any acyclic stage on any graph topology will have its own set of possible dynamics, like described in [76, 77]. Figure 1 illustrates a case of the SER dynamics.

The SMER, Scheduling by Multiple Edge Reversal, is a generalisation of SER where nodes can become sinks more than once inside a periodic cycle. In SMER, any node can have more than one directed edges attached to it (see Fig. 2 for a very simple example of SMER graph dynamics). Also any node is assigned a parameter referred to as its reversibility. Let's randomly take a node, namely node  $i$ , or  $n_i$ , as an example. The reversibility  $r_i$  of node  $n_i$  is defined as the number of the edges directed to  $n_i$  that, when node  $n_i$  is a *sink*, will be released to each of its coupled neighbours

**Fig. 1** Example of the SER graph dynamics. *Black nodes* indicate the sink (active) nodes. Notice that inside the periodic cycle (*Stages 1, 2 and 3*), each node becomes a sink exactly once



**Fig. 2** Example of SMER, the generalised graph dynamics. Nodes  $n_i$  and  $n_j$  have reversibility values 3 and 1, respectively. Grey colour indicates sink nodes. It is clear that the period of this graph dynamics

system has four stages. Node  $n_i$  becomes a sink exactly once, and node  $n_j$  becomes a sink three times inside the period

at the end of  $n_i$  sinking by reversing the direction of  $r_i$  edges to all of its neighbours. Immediately after this reversing operation  $n_i$  becomes a source. The definition of reversibility also implies that any node,  $n_i$ , in a SMER dynamics requires at least the number of  $r_i$  edges directed to it in order to become a sink from a source. In the SER dynamics, all nodes are supposed to have reversibility equal to one. This is because any two neighbours only share one directed edge.

In a mathematically formal description, suppose we have two nodes  $n_i$  and  $n_j$ , where  $n_i, n_j \in N$ , and they are connected with each other. The reversibilities of nodes  $n_i$  and  $n_j$  are  $r_i$  and  $r_j$ , respectively. We have the following rules to choose the number of edges and their directions to ensure the correct operation of SMER [78, 79].

1.  $\max\{r_i, r_j\} \leq e_{ij} \leq r_i + r_j - 1$ , where  $e_{ij}$  is the number of edges between node  $n_i$  and  $n_j$ ;
2.  $f_{ij} = r_i + r_j - \text{gcd}(r_i, r_j)$ . where gcd stands for greatest common divisor, and  $f_{ij}$  is the sum of the greatest multiples of  $\text{gcd}(r_i, r_j)$  that do not exceed the number of edges directed from  $n_i$  to  $n_j$ , and from  $n_j$  to  $n_i$ , respectively, at the initial stage of the graph dynamics.

The first rule stipulates a range for the number of the edges between two coupled nodes, while the second further decides the exact number of edges in the range and their directions. Based on the two rules a dynamic attractor can be made with a flexible control of its active patterns and be immune of the system halt due to deadlock or starvation [77, 78]. Figure 2 illustrates an instance of the generalised dynamics.

In the next section, the SMER graph dynamics will be used to build the artificial CPGs by designing and implementing OBB models. It will be shown that an OBB can be designed as either a digital or an analogue circuit depending on different applications. In both types of OBBs composed of pairs of coupled nodes, a node which has possessed required resources is firing, and meanwhile inhibits its coupled counterpart from becoming a sink. In this case the sinks can be seen as firing neurons in an inhibitorily coupled neuronal network.

### 2.2 Discrete OBB modules and properties

Of long-standing interest are questions about rhythm generation in networks of non-oscillatory neurons, where the

driving force is not provided by endogenous pacemaking cells. A simple mechanism for this is based on reciprocal inhibition between neurons, if they exhibit the property of post-inhibitory rebound [26]. The post-inhibitory rebound mechanism [80] is an intrinsic property of many neurons in the central nervous system, which refers to a period of increased neuronal excitability following the cessation of inhibition. It is often included as an element in computational models of neural networks involving mutual inhibition [81, 82]. The mutually exclusive activity between any two nodes coupled under the aforementioned graph dynamics suggests a scheduling scheme that resembles anti-correlated firing activity between inhibitory neurons exhibiting post-inhibitory rebound. The discrete and the analogue versions of the graph dynamics based OBBs can thus be customised for different rhythmic patterns. The discrete OBB modules are built by directly adopting the generalised graph dynamics, and an asymmetric Hopfield-like neuronal network [83–85] is employed for implementing the analogue OBB modules.

Instead of modelling electrophysiological activities of interconnected neurons based on membrane potential functions, an artificial CPGs network with state machine-based OBBs models the collective behaviour of a neuron set. A simple, discrete OBB module is defined to have an  $r_i$ -sink node, namely a node with a reversibility of  $r_i$ , and an  $r_j$ -sink node (the same meaning of the  $r_i$ -sink), sharing the number of  $e_{ij}$  edges (edges will also be referred to as resources in the subsequent text). Two nodes work as two coupled motor neurons (the nerve cells locate in the spinal cord), respectively, with the shared resources signifying the interactive relationship between the two motor neurons that are interconnected. The exchange of the amount of  $r_i$  (or  $r_j$ ) of resources between two nodes results in pre-defined firing frequencies of two motor neurons.

There is clearly a time relation between the pair of coupled nodes. The larger the reversibility value of a node is, the shorter the firing time of that node will be in a cycle (see the example in Fig. 2). A short firing time of a node

means a small duty cycle (a fraction of the active time in the whole period). Therefore, the duty cycle of a node in an OBB module is easily adjustable by setting the reversibility values for the coupled nodes. Meanwhile, the firing phase between two coupled nodes is also configurable by setting an appropriate value for  $f_{ij}$  in the graph dynamics at the start stage.

The state transition of each leg, e.g. from stance to swing, and the corresponding phase relations among different legs are important to simulate the gait model [86]. The phase circulation can be represented by the circulation of the discrete OBB modules. A simple example of how a possible scheme of firing pattern circulation of OBBs modules can simulate the activity envelope of a pair of flexor and extensor motor neurons is shown in Fig. 3.

After an OBB module is constructed and configured for a coupled pair of flexor and extensor neurons, the envelope of each neuron with a pre-defined duty cycle is formed. For the binary activity in a digital circuit, only a few of sampling time instants are needed in one leg's locomotion period representing the firing of the flexor and extensor neurons, respectively. For a more complex locomotion model, it is the individual OBB modules rather than the integrated OBB network that is governed by the generalised SMER graph dynamics. Therefore, the methodology of constructing a complex locomotion model with the discrete OBB module is fairly straightforward and useful for implementing a modular circuit for the asynchronous operations of a gait model.

### 2.3 Analogue OBB modules and properties

The aforementioned two sorts of graph dynamics have the potential to provide the greatest concurrency among scheduling schemes on resource-sharing systems [76, 79]. The mutually exclusive characteristic between any two coupled nodes makes the scheduling scheme suitably tailored for simulating post-inhibitory rebound, a mechanism widely employed for locomotion and other rhythmic



**Fig. 3** Mimicking rhythmic patterns generated by CPGs with the SMER dynamics. **a** Activity of simplified flexor (node  $n_j$ ) and extensor (node  $n_i$ ) motor neurons during walking of a cockroach and the graph dynamics simulation of its envelope. Here two directed

edges are equivalently replaced by two small, *black circles* on an undirected link. The attaching of a *small circle* to a node represents its direction. **b** The biological signals are reproduced by courtesy of Getting [87]

activities [82, 88]. The generalised SMER graph dynamics depends on the initial allocation of shared resources. Different configurations of shared resources lead to different cyclic behaviours, and even to system deadlock or node starvation if shared resources are not allocated properly. This motivates investigation on how to mimic CPGs with the graph dynamics to simulate numerous rhythmic patterns while avoiding undesirable situations.

It is clear that using graph dynamics for simulating different rhythmic patterns is essentially intuitive due largely to the discrete nature of the dynamics. However, analogue behaviours are ubiquitous in the real world. They are continuous rather than discrete in the time domain, better described by analogue circuitry rather than digital one. Following the digital version, we present a novel continuous time OBB structure that is similar to asymmetric Hopfield networks [83, 84] but governed by the graph dynamics. This structure has been classified into two major categories, namely simple OBB and composite OBB, depending on the network complexity. The work also provides a general mathematical framework for describing the dynamic properties of analogue OBB modules.

Like the dynamics of cellular neural networks [89, 90], the input and output voltages of each node in an OBB module are normalised to the digital low or high level, while the internal potential is continuous within the normalised interval [0, 1]. The simple OBB modules consist of two interconnected nodes with pre-specified reversibilities. The composite OBBs can have an arbitrary number of cells interconnected in any topology. Both types of OBB modules operate under the mechanism of the generalised SMER graph dynamics in which the initial resources configuration is important to avoid possible abnormal situations like deadlock or starvation during operation.

### 2.4 The simple OBB module

Suppose we have a graph represented by  $G(N,E)$ , where  $N$  is a set of nodes and  $E$  the set of edges connecting these nodes in the graph  $G$ , now consider a subgraph of  $G(N,E)$ , namely  $G^{ij}$ , having a pair of coupled nodes  $n_i$  and  $n_j$  with  $r_i$  and  $r_j$  as their reversibility, respectively. This simple network can be translated into a simple OBB module, with the nodes being mapped to the neurons (hence in this work we use nodes or neurons for the same meaning), and the resources being mapped to the synapses and associated weights. The membrane potential of neuron  $n_i$  at time  $t$ ,  $M_i(t)$ , is supposed to depend on three factors, i.e. the potential at last instant  $M_i(t - 1)$ , the impact of its coupled neuron  $n_j$  output  $v_j(t - 1)$ , and the negative feedback of neuron  $n_i$  itself  $v_i(t - 1)$ , without considering the external impulses. Figure 4 shows the circuit representation of the

module. The selection of system parameters, such as the neuron thresholds and synapse weights, are crucial for modelling the OBB module. In the model, let  $r' = h(r)$ , where  $h$  is a function of getting the highest integer scale in  $(r_i, r_j)$ , and then multiplying it by 10, e.g. if  $r_i = 77$  and  $r_j = 463$ , the highest integer scale is 100 from  $r_j$ , and then multiplying it by 10 we have 1000.

So  $h(r) = h(\max(77, 463)) = h(463) = 10^3$ . The neurons' membrane thresholds  $\theta_i$  and  $\theta_j$  and their synaptic weights can be designed as follows,

$$\begin{cases} \theta_i = \frac{r_i}{f_{ij}} \\ \theta_j = 1 - \theta_i \\ w_{ij} = \frac{r_i}{r'} \\ w_{ji} = \frac{r_j}{r'} \end{cases} \tag{1}$$

The system parameters can be arranged by comparing the two nodes' reversibilities. If  $r_i > r_j$ , then we have  $\theta_i > \theta_j$  and  $w_{ij} > w_{ji}$  (i.e. asymmetric coupling), that means a node with smaller reversibility, corresponding to a neuron with lower threshold in an OBB module, will oscillate at a higher frequency than its companion does. It will also weakly stimulate its coupled neuron by contributing with a smaller weight change.  $f_{ij}$  has the same meaning as in the SMER algorithm, and we have  $\theta_j = \frac{f_{ij} - r_i}{f_{ij}} = 1 - \frac{r_i}{f_{ij}} = 1 - \theta_i$ . This arrangement of model parameters ensures that the behaviour of the SMER-based, analogue OBB modules is isomorphic with the behaviour of the original SMER algorithm. The difference equation in the discrete time domain of this system can be formulated as follows: Each neuron's self-feedback strength is  $w_{ii} = -w_{ij}$ ,  $w_{jj} = -w_{ji}$ , respectively, and the activation function, which reflects the neuron's output in relation of its model parameters, is a sigmoidal Heaviside type. Thus, we have,

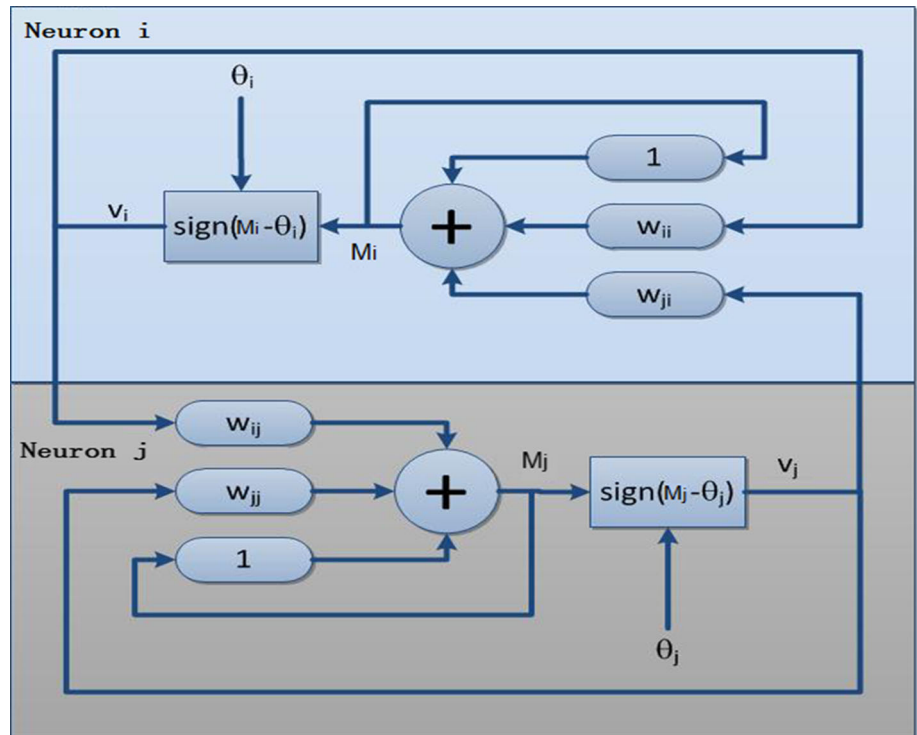
$$\begin{bmatrix} M_i(t) \\ M_j(t) \end{bmatrix} = \begin{bmatrix} M_i(t-1) \\ M_j(t-1) \end{bmatrix} + \frac{1}{r'} \times \underbrace{\begin{bmatrix} -r_i & r_j \\ r_i & -r_j \end{bmatrix}}_W \times \begin{bmatrix} v_i(t-1) \\ v_j(t-1) \end{bmatrix} \tag{2}$$

where  $W$  is the weight matrix. We have the outputs of neurons as,

$$\begin{cases} v_i(t) = \max(0, \text{sgn}(M_i(t) - \theta_i)) \\ v_j(t) = \max(0, \text{sgn}(M_j(t) - \theta_j)) \end{cases} \tag{3}$$

The designed circuit can be considered as a conservative dynamical system in an ideal case, i.e. the total energy is constant. The sum of two neurons' membrane potentials at any given time is normalised to one. It will be shown that the system is able to develop from an arbitrary initial state into a limit cycle (for an oscillatory period), with the firing rate of each neuron adjustable based on its reversibility.

**Fig. 4** Diagram of a simple OBB module as the basic oscillation unit



However, like most dynamic systems, this model has a limitation in its dynamic range. There exists a singular point when each neuron’s membrane potential equals to its threshold. In this case, Eq. (3) becomes  $v_i(n) = v_j(n) = 0$ , hence Eq. (2) accordingly becomes:  $M_i(n) = M_i(n - 1)$  and  $M_j(n) = M_j(n - 1)$ . The system behaviour can be unpredictable: It may transit to a different oscillatory pattern due to small external perturbations, or halt with no perturbations. When designing an oscillatory neural network using OBB modules, it is possible to avoid the occurrence of any singular point state by presetting the initial membrane potential  $M_i(0)$  to an appropriate value so that  $M_i(0) \neq \theta_i$ . Within the normal dynamic range except for the singular point, the circuit of the OBB module represents a starvation- and deadlock-free dynamic system, and it is a stable and periodically oscillating system no matter what initial potentials its neurons may have [75].

The combination of the duty cycles, the oscillation frequencies, and the phase difference of a coupled pair of neurons is the key set of parameters for modelling a joint of one degree of freedom (DOF). The oscillatory pattern transition, which is another important concept in addition to the pattern generation, can thus be understood as a transition from an old to a new set of the joint parameters. It is clear that the duty cycle of an extensor motor neuron plays an important role in deciding the locomotion speed of a legged animal [91–93]. In our model, the duty cycle of a neuron in a coupled, two-neuron system is dependent on the model parameters shown in Eq. (1). The choice of

reversibilities of two coupled nodes thus dictates the transition between different patterns as it decides the system parameters, and hence the duty cycles. Therefore, in our study the design of transition between patterns is simplified to the selection among different reversibility values of two coupled nodes. On the other hand, the proposed pattern generation model has a linear internal dynamics, i.e. the membrane potential is a linear function of three input variables, and a nonlinear, Heaviside-type dynamics in the output of a neuron as a function of the membrane potential and the threshold of that neuron.

Suppose both coupled nodes have their reversibility changed in an amount of  $r_i^d$  and  $r_j^d$ , respectively, we can rewrite formula (1) to obtain the model parameters for the new pattern, as shown below.

$$\begin{cases} \theta_i^{\text{new}} = \frac{r_i + r_i^d}{f_{ij}^{\text{new}}} \\ \theta_j^{\text{new}} = \frac{r_j + r_j^d}{f_{ij}^{\text{new}}} \\ w_{ij}^{\text{new}} = \frac{r_i + r_i^d}{r'} \\ w_{ji}^{\text{new}} = \frac{r_j + r_j^d}{r'} \end{cases} \quad (4)$$

where  $f_{ij}^{\text{new}} = r_i + r_i^d + r_j + r_j^d - \text{gcd}(r_i + r_i^d, r_j + r_j^d)$ . Here we make a reasonable assumption that  $r'$  is unchanged as the amount of  $r_i^d$  or  $r_j^d$  is in the similar scale of its old

value, respectively. This ensures that the originally linear membrane potential dynamics of a neuron is still linear after pattern transition. We further assume that, according to SMER, a firing neuron will releases some shared resources in the amount equal to its new reversibility to its coupled neuron, which receives them passively and will possibly become able to fire in one of the subsequent stages. The firing neuron in the dynamic model will thus decrease in its membrane potential by an amount equal to the new strength of its output synapse, while the coupled neuron will have the same amount added to its membrane potential. We can rewrite Eq. (2) for the membrane dynamics in a more general format involving pattern transition mechanism as follows.

$$\begin{aligned}
 \begin{bmatrix} M_i(n) \\ M_j(n) \end{bmatrix} &= \begin{bmatrix} M_i(n-1) \\ M_j(n-1) \end{bmatrix} \\
 &+ \left( W + \underbrace{\frac{s_i}{r'} \begin{bmatrix} -r_i^d & 0 \\ r_i^d & 0 \end{bmatrix} + \frac{s_j}{r'} \begin{bmatrix} 0 & r_j^d \\ 0 & -r_j^d \end{bmatrix}}_{\Delta W} \right) \\
 &\times \begin{bmatrix} v_i(n-1) \\ v_j(n-1) \end{bmatrix} \tag{5}
 \end{aligned}$$

where,

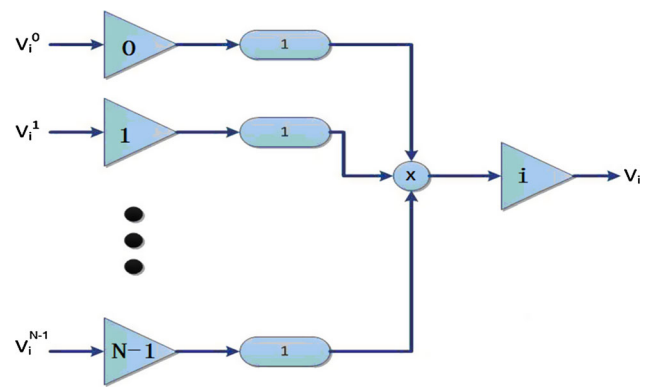
$$s_k = \begin{cases} 1, & \text{if } r_k \text{ changed} \\ 0, & \text{if } r_k \text{ not changed} \end{cases} \tag{6}$$

is a control signal for pattern transition, here  $k \in (i, j)$ . This equation indicates that, in theory, the pattern transition can be induced by the reversibility change of any one of the two coupled neurons.

### 2.5 The composite OBB module

A composite OBB module is a generalisation of the simple OBB module which consists of an arbitrary number of simple OBB modules. A composite OBB can have more than two neurons and therefore, a more complex topology. Suppose a graph  $M(N, E)$  containing a set of  $m$  neurons in a set  $N = \{n_1, n_2, \dots, n_m\}$ , and a set of connections  $E = \langle 0 | 1 \rangle_{ij}$ , where  $\forall i, j \in N$ , which defines the connection topology of this graph by using  $\langle 1 \rangle_{ij}$  and  $\langle 0 \rangle_{ij}$  to indicate the presence or absence of a connection between nodes  $i$  and  $j$ , respectively. Node  $i$  has its reversibility  $r_i$ . There are  $e_{ij}$  shared resources on the corresponding connection  $\langle 1 \rangle_{ij}$ , with their number and configuration stipulated by the rules set in the SMER section.

Different from a simple OBB module composed by exactly two coupled neurons, a composite OBB module has at least one neuron connected to at least two other neurons.



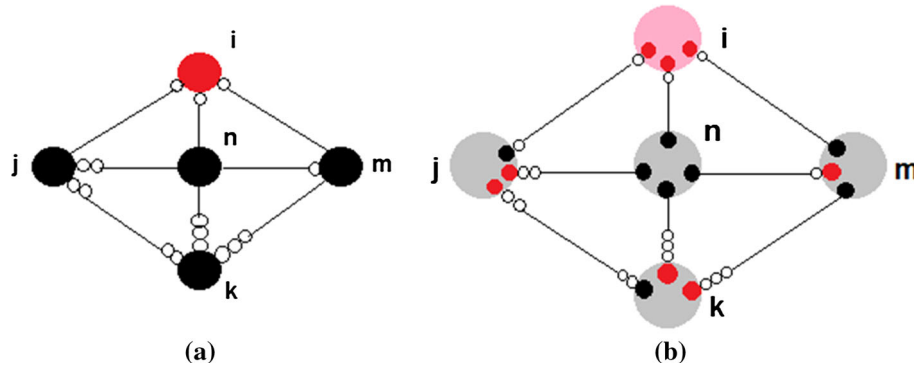
**Fig. 5** Diagram of one macroneuron (represented by the right triangle) and its clones (represented by a column of the left triangles) in a composite OBB module

Figure 5 shows that a neuron  $i$  has connections to a total number of  $N$  neurons. The composite OBB module can be dissected into various simple OBB modules. For instance, in Fig. 5, neuron  $i$  can be split into  $N - 1$  copies, which share the same local clock of their maternal neuron  $i$ , to connect to the  $N$  input neurons. Here, a copy of neuron  $i$ , plus the corresponding input neuron and their connection, forms a simple OBB module. We terminologically regard the output neuron (like neuron  $i$  in Fig. 5) in a composite OBB module as a macroneuron and macroneuron’s copies as its clones.

According to SMER, a macroneuron of the composite OBB module operates in a “whole-or-none” mechanism in terms of the activity of its clones. The macroneuron is active if and only if all of its clones are active. Fundamentally, a composite OBB module operates based on its constituent simple OBB modules. A schematic diagram of an exemplar composite OBB module and its equivalent representation is shown in Fig. 6.

As a generalised version, a composite OBB module can represent a more complex oscillatory neuronal network and flexibly reproduce more complex cycles of rhythmic patterns than a simple OBB module. A composite OBB module is formed based on any prescribed oscillatory functionality in a specific application, so it does not have a fixed form in terms of the number of constituent clones, the connection structure, etc. Different applications will specify different macroneurons and their topology, and subsequently use them as the oscillatory building blocks. Due to the fact that a composite OBB module is dependent on specific applications and impossible to know beforehand, a rule of thumb to analyse this kind of modules is to dissect them into the subsystems of simple OBB modules where Eqs. (1)–(6) apply.

Since the output of a macroneuron  $i$  is determined by all its clones, we have,



**Fig. 6** Two equivalent architectures to illustrate the SMER algorithm in an exemplar composite OBB. Here the reversibility values  $r_i = r_m = r_n = 1$ ;  $r_j = 2$ ;  $r_k = 3$ . **a** One operation stage of SMER in a composite OBB module, the node in red is active, those in black are inactive, the *small circles* represent resources. **b** An alternative, equivalent description for the stage of the composite OBB module

$$V_i(k) = \prod_{j=1}^n v_i^j(k) \tag{7}$$

where  $\forall i \in N, i \neq j$  and  $v_i^j$  is the output of clone  $j$  of the macroneuron  $i$ , which is linked to a corresponding clone of another macroneuron  $j$  via the connection  $\langle 1 \rangle_{ij}$ . The superscript sequence  $j = 1, 2, \dots, n$  is the clone number of a macroneuron  $n_i$ , which also represents for the other macroneurons to which macroneuron  $i$  connects.

It is also noticeable about how to choose the initial membrane potential values for the clones which are parts of the simple OBB modules. Different choices will lead to different initial self-organisation process of a composite OBB module. To avoid system halt, no clone should be inactive if its macroneuron is designed to be active by Eq. (7). Within an appropriate parameter range (i.e. a neuron’s membrane potential is not equal to its threshold), a random selection of initial membrane potential values is allowed. After a possible initial duration whose length is determined by the choice of initial membrane potentials, the system will oscillate periodically.

### 3 Primer of computer simulation

In this section, the operations of the OBB modules are preliminarily demonstrated through several computer simulated works. Since a composite OBB module is fundamentally composed by some simple OBB modules (see Fig. 6), it will have the collective behaviours of the simple OBB modules. Therefore, the experiments focus on the operations of the simple OBB modules by showing some case studies of their dynamics in terms of the oscillatory patterns generation and transition.

with the macroneurons and their clones, the *pink node* is active with all of its clones being active; the *grey nodes* are inactive with at least one of its clones being inactive. It is clear that the independent operations of eight simple OBB modules (represented by eight connections and their associated clones, and resources) decide the operation of this composite OBB module (colour figure online)

### 3.1 Pattern generation

Let us suppose that a simple OBB module has two coupled neurons  $i$  and  $j$  with the reversibility values  $r_i = 3$  and  $r_j = 12$ . According to the theory, we have the module parameters as follows.

$$\begin{cases} \theta_i = 0.25 \\ \theta_j = 0.75 \\ w_{ij} = 0.03 \\ w_{ji} = 0.12 \end{cases}$$

If we further assume that the initial membrane potentials of two neurons are  $M_i(0) = 0.66$  and  $M_j(0) = 0.34$ , respectively, we can obtain the oscillatory dynamics of this simple OBB system by using MATLAB–Simulink, as shown in Fig. 7.

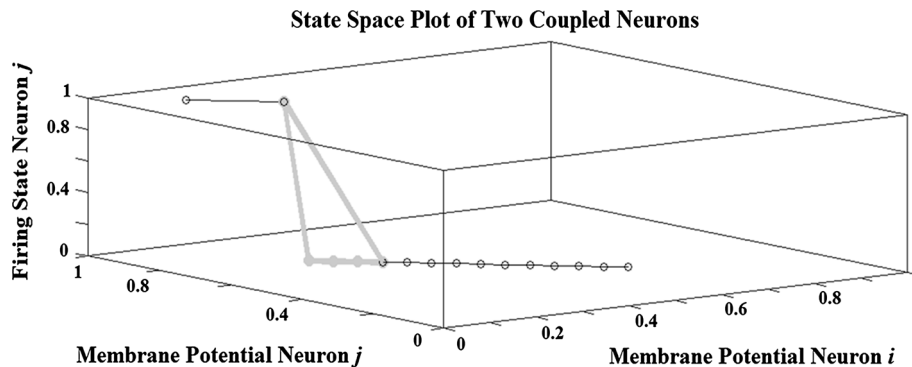
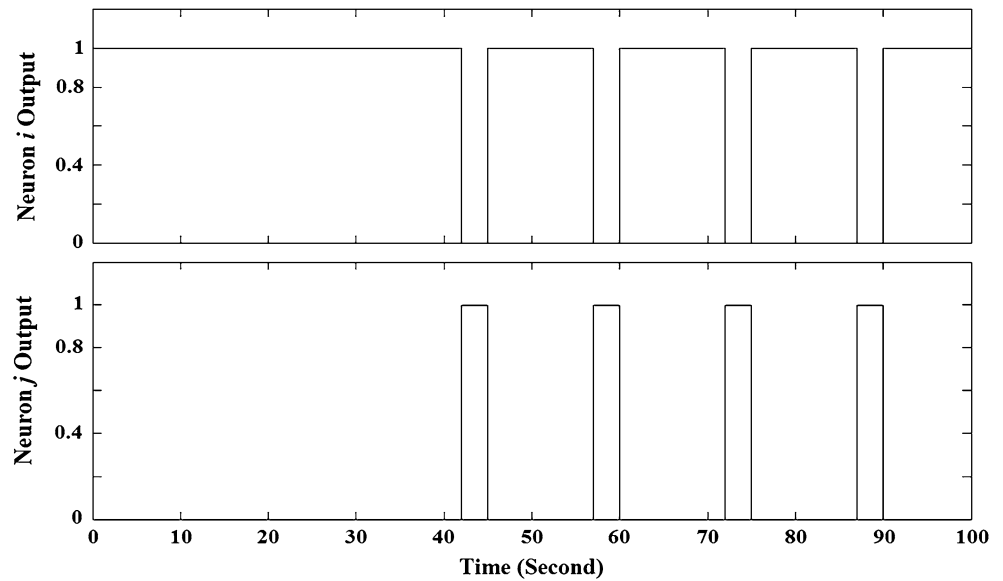
It is noticeable that the coupled neurons start with a self-organised period with the given initial membrane potentials. Then the system undergoes a stable periodic oscillation. The duty cycle of a neuron is decided by the model parameters, and thus indirectly related with the reversibilities of two coupled neurons. The state space plot of this example is shown in Fig. 8.

Given the initial membrane potentials of two coupled neurons in the model as  $M_i(0) = 0.66, M_j(0) = 0.34$ , the state of the model develops through an initial stage into a limit cycle (as shown in Fig. 8 in thick and grey line), which corresponds to a stable periodic oscillation in the time domain (as shown in Fig. 7). Another example of choice of the initial membrane potentials of the coupled neurons, e.g.  $M_i(0) = 0.12, M_j(0) = 0.88$ , results in a shorter journey before converging to the limit cycle (see Fig. 8).

It is clear that neuron  $j$  is active for a much shorter time duration than neuron  $i$ , which becomes active in the rest of



**Fig. 7** Simulated dynamics of a simple OBB module in time domain. *Upper panel:* neuron *i* output; *Lower panel:* neuron *j* output. When the system becomes stable, the oscillatory period is 15 s and the duty cycle of neuron *i* is 12 s. Note that here (and in the subsequent simulations) second is used as the nominal time unit for illustrative purpose only. In practice the time unit is closely related to the units of the system clock, and should be conditioned to any suitable unit size



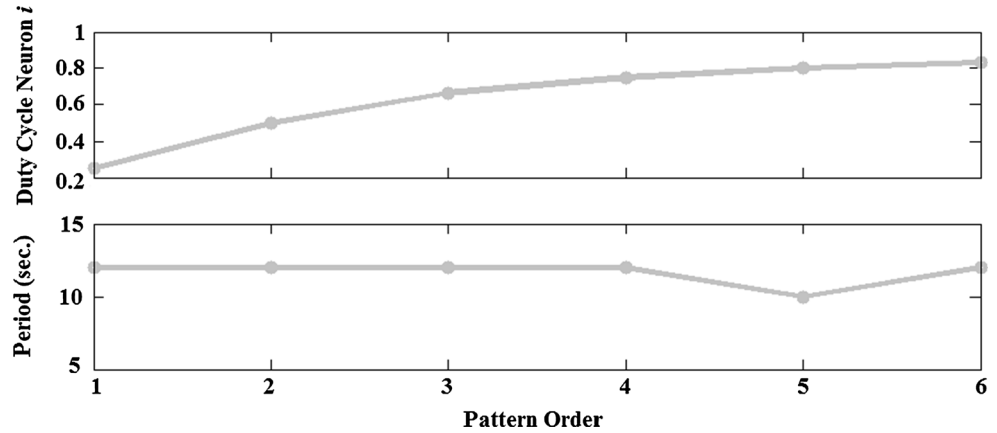
**Fig. 8** State space plot of the periodic oscillation of two coupled neurons under the graph dynamics. The two axes on the planar surface represent the membrane potentials of two neurons, and the vertical

axis is for the firing state of neuron *j*. From different initial membrane potentials, the model evolves into a same sequence of periodical states like a limit cycle

the time when neuron *j* is inactive. The duty cycle of neuron *i* is calculated as  $\frac{4}{5}$  for a period of 15 s (Fig. 7). If we fix the reversibility value  $r_i = 3$ , and let  $r_j \in [1, 3, 6, 9, 12, 15]$ , we can get the duty cycle of neuron *i* in a corresponding array as  $\left[\frac{1}{4}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}\right]$ , and the period of oscillation in an array [6, 9, 12, 15, 18]. Without loss of generality we can use frequency-conditioning techniques to round the period of oscillation of the OBB module to be as close as possible to a same value (here 12), while keeping the duty cycle of two neurons unchanged so that, in applications, we can compare different patterns generated with a same baseline. A new array of oscillation periods becomes [10, 12]. A plot of the duty cycle of neuron *i* and the period of oscillation of the model is shown in Fig. 9. Following Pearson’s argument [91] that a faster gait pattern corresponds to a larger duty cycle for swing

phase and therefore a smaller duty cycle for stance phase, it is possible to vary the locomotion speed by adjusting the reversibilities of two coupled neurons (corresponding to the swing and stance phases, respectively) in a graph dynamics driven system. When the duty cycles of two coupled neurons are changed by using different reversibilities, the model period remains largely unchanged. This phenomenon displayed by the model is consistent with that of our previous study in implementing a nonlinear stepping pattern generation model with analogue circuitry [58], and therefore the current model provides an alternative approach to modelling the CPG mechanism amenable to circuit realisation, due largely to its characteristics of modularity and scalability. The OBB circuit is event driven and made of the discrete building blocks. These features facilitate its implementation in asynchronous sequential logic circuits.

**Fig. 9** Duty cycle and oscillation period of a simple OBB module with a fixed reversibility  $r_i = 3$ , and variable reversibility  $r_j$  as shown in the text. The pattern order corresponds to the normalised oscillation periods [10, 12]



The continuous-valued version of the SMER algorithm can be used to design a building block for generating rhythmic patterns with different duty cycle values for the activity of a coupled neurons in a flexible and systematic approach. For instance, if one more duty cycle point is needed between the values of 1/4 and 1/2, say 2/5, a composite OBB module can be designed for this additional state with the new duty cycle (see Fig. 10).

In this design, a composite OBB has three macroneurons labelled as  $k$ ,  $m$ , and  $n$ . Each macroneuron has two clones inside. It is clear that this composite OBB module is composed of three simple OBB modules, which are clones and resources connecting macroneurons  $k$  and  $m$ ,  $k$  and  $n$ ,  $m$  and  $n$ , respectively. The clones inside macroneuron  $k$  are referred to as  $C_{km}^k$  and  $C_{kn}^k$ , in which the subscripts represent the connections between different neuron pairs of  $k$ – $m$  and  $k$ – $n$ , the superscripts denote the clones are in macroneuron  $k$ . The reversibility variables of these two clones are labelled and set as  $r_{km}^k = 1$  and  $r_{kn}^k = 2$ , respectively. Similarly, the clones in macroneuron  $m$  are  $C_{mk}^m$  and  $C_{mn}^m$ , their reversibility variables are  $r_{mk}^m = 4$  and  $r_{mn}^m = 1$ . The clones in macroneuron  $n$  are  $C_{nk}^n$  and  $C_{nm}^n$ , their reversibility variables are  $r_{nk}^n = 3$  and  $r_{nm}^n = 1$ . The model parameters of the simple OBB modules connecting macroneuron  $k$ ,  $m$ , and  $n$ , respectively, can be derived as follows.

$$\begin{cases} \theta_{km}^k = 0.25 \\ \theta_{km}^m = 0.75 \\ w_{km}^k = 0.1 \\ w_{km}^m = 0.4 \end{cases} \quad \begin{cases} \theta_{kn}^k = 0.5 \\ \theta_{kn}^n = 0.5 \\ w_{kn}^k = 0.2 \\ w_{kn}^n = 0.3 \end{cases} \quad \begin{cases} \theta_{nm}^n = 0.5 \\ \theta_{nm}^m = 0.5 \\ w_{nm}^n = 0.1 \\ w_{nm}^m = 0.1 \end{cases}$$

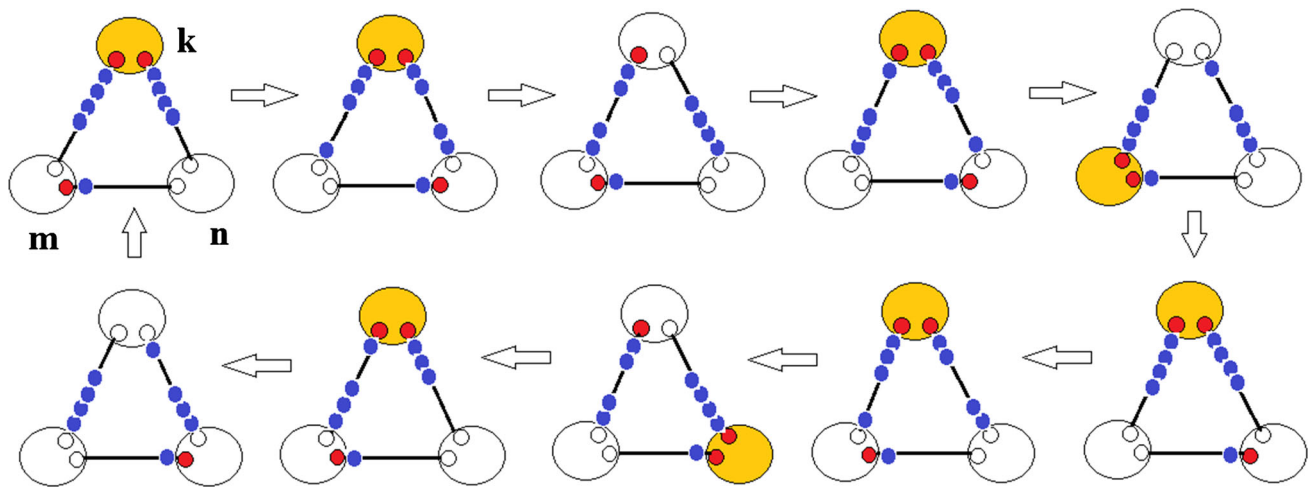
After selecting the model parameters by computing OBB parameters based on SMER, the initial membrane potentials for the coupled clones in the simple OBB modules can be chosen randomly with the range [0, 1], as it is aforementioned that the random selection of the initial membrane potentials will possibly lead to an initial self-

organisation stage whose length is dependent on the initial potential values. As long as the singular values of the membrane potentials are avoided, the OBB will reach a limit cycle. In this example, macroneuron  $k$  is expected to fire for two consecutive stages in the five-stage cycle for obtaining the pre-defined duty cycle. According to Eq. (7), the states of its two clones will be multiplied to have the target output. Here the initial membrane potentials of the coupled neurons in three simple OBB modules can be set as:  $M_{km}^k(0) = 0.5$ ,  $M_{km}^m(0) = 0.5$ ,  $M_{kn}^k(0) = 0.75$ ,  $M_{kn}^n(0) = 0.25$ ,  $M_{nm}^n(0) = 0.75$ ,  $M_{nm}^m(0) = 0.25$ . The simulation outcome of this continuous-valued composite OBB module using MATLAB–Simulink is shown in Fig. 11.

### 3.2 Pattern transition

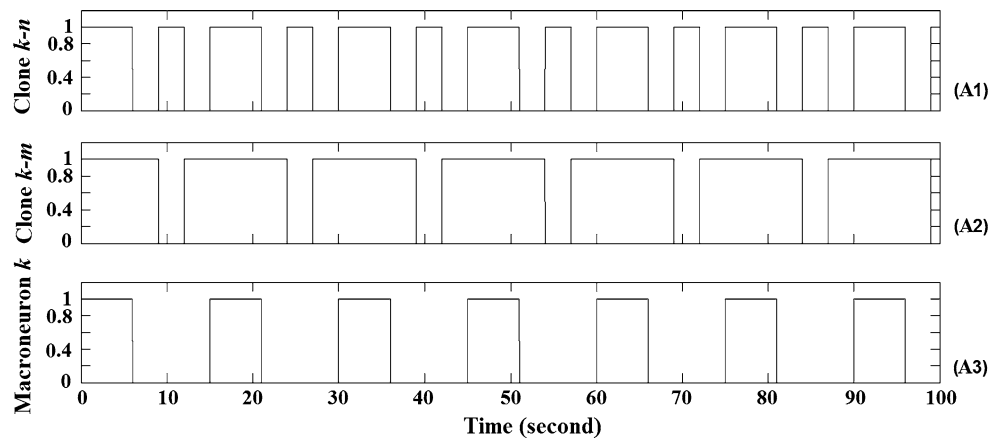
As we have shown in Eq. (5), a change in the reversibility of any one of two coupled neurons will result in the change of model parameters, hence the change of oscillatory patterns. Therefore, the pattern transition in the OBB module-based CPG architecture is straightforward. In a Simulink simulation, one is able to use one time-related control signal, corresponding to the control variable in Eq. (6), to switch between the old and the new model parameters, derived from the old and the new reversibilities of coupled neurons. For instance, if one needs to change the reversibility of a pair of coupled neurons from  $\{r_i = 3, r_j = 12\}$  to  $\{r_i = 3, r_j = 3\}$ , the dynamic model parameters are changed accordingly, like a switch being used to control this change. A transition between the old and the new patterns can be achieved with some possible intermediate self-organisation period (see Fig. 12).

$$\begin{cases} \theta_i = 0.25 \\ \theta_j = 0.75 \\ w_{ij} = 0.03 \\ w_{ji} = 0.12 \end{cases} \Rightarrow \begin{cases} \theta_i = 1 \\ \theta_j = 0 \\ w_{ij} = 0.3 \\ w_{ji} = 0.3 \end{cases}$$



**Fig. 10** Exemplar composite OBB module containing 3 macroneurons to achieve a new duty cycle for a macroneuron *k*. The rest two macroneurons are used to support the design. The red colour represents for active clones while the yellow colour for active macroneurons. The shared resources are small blue circles (colour figure online)

**Fig. 11** Simulation of a composite OBB designed to show a firing duty cycle of 0.4 for a macroneuron. (A1) and (A2) show the output states of two clones of macroneuron *k*, respectively. (A3) is obtained through Eq. (7), the output of the macroneuron *k*, which meets the design requirement in duty cycle

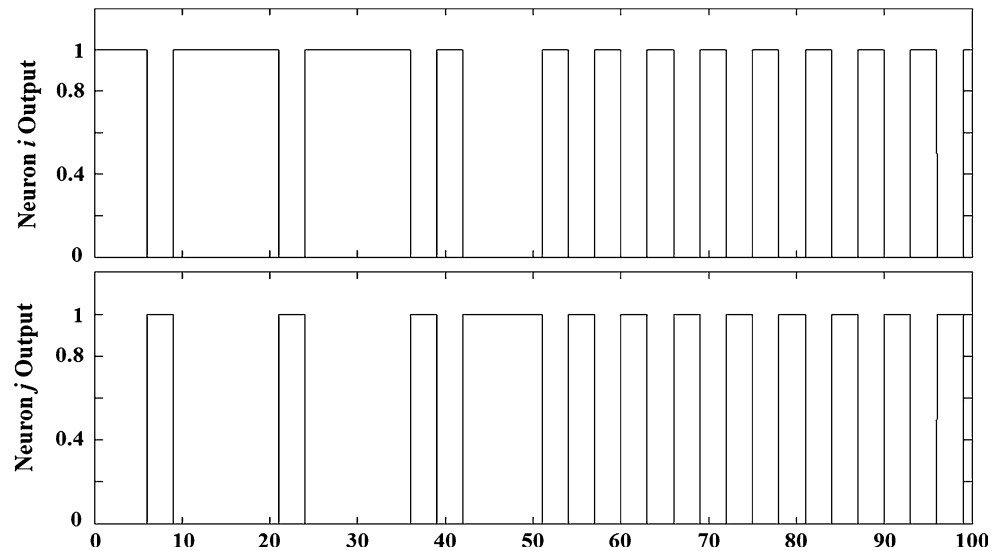


It is clear that, if no transition happens, then neuron *i* will continue its first pattern, which becomes high at time instant 39 and lasts for 12 s till 51. The duty cycle for neuron *i* is 0.8 (and for neuron *j* is 0.2 accordingly). As a command for pattern transition occurs at time instant 40, ideally the new pattern should start immediately after this time instant. Practically a self-organisation cycle exists such that the new pattern starts at the time instant 51. This is because membrane potentials of the two coupled neurons are not ready (or, not as close as possible to their thresholds due to the operation of the old pattern) to make the transition happen immediately. After a short period, though, the model will evolve into the new designed pattern with the duty cycle of neuron *i* as 0.5 (neuron *j* as 0.5). It is suggested that this phenomenon be biologically plausible as no real creatures will act immediately, i.e. without latency, upon a command of action.

### 4 Discussion and conclusion

Although most CPG models proposed so far are mainly imaginary architectures with no clear neurophysiological proofs, advances on neuroscience start to reveal the existence of this widely accepted mechanism behind animal locomotion. Recently Hückesfeld et al. [94] identified the distinct clusters of motor neurons responsible for certain types of motions, and the underlying CPG mechanisms. On the other hand, the recent applications of CPG models for controlling robots are still largely following the tradition of using dynamical system methods to formulate the periodic oscillations represented by, e.g. limit cycles (for a review see [95]). Some new trends, however, are to incorporate the interactions of CPGs and the external world by using feedback signals from sensors, or to drive different joints (degrees of freedom) on animal legs rather than the

**Fig. 12** Oscillatory pattern transition process. Transition occurs at the time instant 40; before that reversibilities are  $\{r_i = 3, r_j = 12\}$ ; afterwards  $\{r_i = 3, r_j = 3\}$ .



legendary, more abstract phase relations between legs by ignoring the control of intra-leg joints [96–98]. Furthermore, apart from the more classic CPG models such as the van der Pol and Wilson–Cowan oscillators, new oscillators, e.g. the Morphed oscillator [99] which exhibits arbitrary limit cycle shapes and can represent first, second or  $n$ th order dynamical systems, are proposed and used to build a CPG model with the ameliorated performance like perturbation resistance [100]. Despite the mathematical soundness of these works, however, most of their effects on controlling legged robots are demonstrated by using computer simulations. Much more works remain if these models are to be implemented in hardware for autonomous control of a robot.

Recent neuromorphic hardware realisation of the CPG models has shown great potential in implementing silicon neurons and their networks to control the robot behaviours [52–60]. Nevertheless, many of the recent works are still focused on the basis of building the neural oscillators, showing their characteristics, rather than using the oscillators to construct a specific CPG architecture [101, 102]. This is possibly due to the lack of a modular and scalable approach to designing a CPG architecture by using a range of the available building blocks.

A novel oscillatory building block (OBB) model that is able to be configured to create a tailor designed architecture for both locomotion rhythmic pattern generation and transition has been proposed in this work. Since the simple OBB module is the basis from which a rhythm-producing model of high complexity can be designed, the top-level model is highly modular and scalable for design, prototype, manufacture and test. It is also an asynchronous and self-clocked system if the reversibility values and initial membrane potentials are chosen for individual simple

OBBs. Because of the simplicity of the system, a hardware version of a simple OBB module can be made in such a way that, it is possible to develop a real-time hardware implementation of systems with arbitrary complexity.

The resulting continuous, linear and time-invariant system has its correctness based on the SMER graph dynamics. Therefore, a smooth operation of the designed workflow of rhythmic patterns is ensured. An intermediate self-organisation period may exist but this phenomenon is not an issue, being compatible with its biological counterparts. Pattern transition is controlled by just one model parameter, i.e. the reversibility of a neuron, in order to update the gait pattern according to the environment. Future works include the design of OBB architectures for a pre-specified, legged animal species, in order to show the model's capability on mimicking a variety of legged locomotion patterns in a flexible and systematic approach.

**Acknowledgments** This research has been supported in part by a British Biotechnology and Biological Sciences Research Council (BBSRC) Grant BBS/B/07217, an Engineering and Physical Sciences Research Council (EPSRC) Grant EP/E063322/1, and two Chinese National Science Foundation (NSFC) Grants 60673102 and 61103185. The MATLAB–Simulink program is available upon request.

## References

1. Büschges A (2005) Sensory control and organization of neural networks mediating coordination of multisegmental organs for locomotion. *J Neurophysiol* 93(3):1127–1135
2. Grillner S (2006) Biological pattern generation: the cellular and computational logic of networks in motion. *Neuron* 52:751–766
3. Cruse H, Dürr V, Schilling M, Schmitz J (2009) Principles of insect locomotion. In: Arena P, Patanè L (eds) Spatial temporal patterns for action-oriented perception in roving robots, cognitive systems monographs. Springer, Berlin, pp 43–96

4. Mulloney B, Smarandache C (2010) Fifty years of CPGs: two neuroethological papers that shaped the course of neuroscience. *Front Behav Neurosci* 4:45. doi:10.3389/fnbeh.2010.00045
5. Fuchs E, Holmes P, Kiemel T, Ayali A (2011) Inersegmental coordination of cockroach locomotion: adaptive control of centrally coupled pattern generator circuits. *Front Neural Circuits* 4:125. doi:10.3389/fncir.2010.00125
6. Grillner S (1975) Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiol Rev* 55:247–304
7. Grillner S (1985) Neurobiological bases of rhythmic motor acts in vertebrates. *Science* 228:143–149
8. Stein PSG (1978) Motor systems with specific reference to the control of locomotion. *Annu Rev Neurosci* 1:61–81
9. Schöner G, Kelso JA (1988) Dynamic pattern generation in behavioural and neural systems. *Science* 239:1513–1520
10. Alexander RM (1989) Optimization and gaits in the locomotion of vertebrates. *Physiol Rev* 69:1199–1227
11. Cohen AH (1992) The role of heterarchical control in the evolution of central pattern generators. *Brain Behav Evol* 40:112–124
12. Pearson KG (1993) Common principles of motor control in vertebrates and invertebrates. *Annu Rev Neurosci* 16:265–297
13. Latash ML (1993) Control of human movement. Human Kinetics Publishers, Champaign
14. McGeer T (1993) Dynamics and control of bipedal locomotion. *J Theor Biol* 163:277–314
15. Ryckebusch S, Wehr M, Laurent G (1994) Distinct rhythmic locomotor patterns can be generated by a simple adaptive neural circuit: biology, simulation, and VLSI implementation. *J Comput Neurosci* 1(4):339–358
16. Kiehn O, Butt SJ (2003) Physiological, anatomical and genetic identification of CPG neurons in the developing mammalian spinal cord. *Prog Neurobiol* 70:347–361
17. Marder E, Calabrese RL (1996) Principles of rhythmic motor pattern generation. *Physiol Rev* 76:687–717
18. Stein PSG, Grillner S, Selverston AI, Stuart DG (eds) (1997) Neurons, networks, and motor behavior. MIT Press, Cambridge
19. Schmidt RA, Lee TD (1999) Motor control and learning: a behavioral emphasis. Human Kinetics Publishers, Champaign
20. Pearson K, Gordon J (2000) Locomotion. In: Kandel E, Schwartz J, Jessel T (eds) Principles of neural science. McGraw-Hill Companies Inc, New York, pp 737–755
21. Magill RA (2001) Motor learning: concepts and applications. McGraw-Hill Companies Inc, New York
22. Grillner S (2003) The motor infrastructure: from ion channels to neuronal networks. *Nat Rev Neurosci* 4:573–586
23. Jing J, Cropper EC, Hurwitz I, Weiss KR (2004) The construction of movement with behavior-specific and behavior-independent modules. *J Neurosci* 24:6315–6325
24. Grillner S, Markram H, De Schutter E, Silberberg G, LeBeau FE (2005) Microcircuits in action—from CPGs to neocortex. *Trends Neurosci* 28:525–533
25. Norris BJ, Weaver AL, Morris LG, Wenning A, García PA, Calabrese RL (2006) A central pattern generator producing alternative outputs: temporal pattern of premotor activity. *J Neurophysiol* 96:309–326
26. Wang XJ, Rinzal J (1992) Alternating and synchronous rhythms in reciprocally inhibitory model neurons. *Neural Comput* 4:84–97
27. Hatsopoulos N (1996) Coupling the neural and physical dynamics in rhythmic movements. *Neural Comput* 8:567–581
28. Rinzal J, Terman D, Wang X, Ermentrout B (1998) Propagating activity patterns in large-scale inhibitory neuronal networks. *Science* 279:1351–1355
29. Ermentrout GB, Chow CC (2002) Modeling neural oscillations. *Physiol Behav* 77:629–633
30. Ghigliazza RM, Holmes P (2004) A minimal model of a central pattern generator and motoneurons for insect locomotion. *SIAM J Appl Dyn Syst* 3(4):671–700
31. Katz PS, Sakurai A, Clemens S, Davis D (2004) Cycle period of a network oscillator is independent of membrane potential and spiking activity in individual central pattern generator neurons. *J Neurophysiol* 92:1904–1917
32. Marder E, Bucher D, Schulz DJ, Taylor AL (2005) Invertebrate central pattern generation moves along. *Curr Biol* 15:R685–R699
33. Jones SR, Kopell N (2006) Local network parameters can affect inter-network phase lags in central pattern generators. *J Math Biol* 52(1):115–140
34. Daun-Gruhn S, Büschges A (2011) From neuron to behavior: dynamic equation-based prediction of biological processes in motor control. *Biol Cybern* 105:71–88
35. Olree KS, Vaughan CL (1995) Fundamental patterns of bilateral muscle activity in human locomotion. *Biol Cybern* 73(5):409–414
36. Prentice SD, Patla AE, Stacey DA (1998) Simple artificial neural network models can generate basic muscle activity patterns for human locomotion at different speeds. *Exp Brain Res* 123(4):474–480
37. Delcomyn F (1999) Walking robots and the central and peripheral control of locomotion in insects. *Auton Robots* 7(3):259–270
38. Ijspeert AJ, Kodjabachian J (1999) Evolution and development of a central pattern generator for the swimming of a Lamprey. *Artif Life* 5(3):247–269
39. Ijspeert AJ (2001) A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biol Cybern* 84(5):331–348
40. McCrear DA (2001) Spinal circuitry of sensorimotor control of locomotion. *J Physiol* 533(1):41–50
41. Butt SJ, Kiehn O (2003) Functional identification of interneurons responsible for left-right coordination of hindlimbs in mammals. *Neuron* 38:953–963
42. Cangiano L, Grillner S (2005) Mechanisms of rhythm generation in a spinal locomotor network deprived of crossed connections: the lamprey hemicord. *J Neurosci* 25:923–935
43. Stein PSG (2005) Neuronal control of turtle hindlimb motor rhythms. *J Comp Physiol A* 191(3):213–229
44. Bucher D, Prinz AA, Marder E (2005) Animal-to-animal variability in motor pattern production in adults and during growth. *J Neurosci* 25:1611–1619
45. Pinto CMA, Golubitsky M (2006) Central pattern generators for bipedal locomotion. *J Math Biol* 53(3):474–489
46. Raibert MH (1986) Legged robots that balance. MIT Press, Cambridge
47. Arena P (2000) The central pattern generator: a paradigm for artificial locomotion. *Soft Comput* 4(4):251–266
48. Buono PL (2001) Models of central pattern generators for quadruped locomotion. *J Math Biol* 42(4):327–346
49. Webb B (2002) Robots in invertebrate neuroscience. *Nature* 417:359–363
50. Cruse H (2002) The functional sense of central oscillations in walking. *Biol Cybern* 86(4):271–280
51. Nusbaum MP, Beenhakker MP (2002) A small-systems approach to motor pattern generation. *Nature* 417:343–350
52. Lewis MA, Etienne-Cummings R, Hartmann MJ, Xu ZR, Cohen AH (2003) An in silico central pattern generator: silicon oscillator, coupling, entrainment, and physical computation. *Biol Cybern* 88(2):137–151
53. Nakada K, Asai T, Amemiya Y (2003) An analog CMOS central pattern generator for interlimb coordination in quadruped. *IEEE Trans Neural Netw* 14(5):1356–1365

54. Still S, Hepp K, Douglas RJ (2006) Neuromorphic walking gait control. *IEEE Trans Neural Netw* 17(2):496–508
55. Vogelstein RJ, Tenore F, Etienne-Cummings R, Lewis MA, Cohen AH (2006) Dynamic control of the central pattern generator for locomotion. *Biol Cybern* 95(6):555–566
56. Zhang X, Zheng H, Chen L (2006) Gait transition for a quadrupedal robot by replacing the gait matrix of a central pattern generator model. *Adv Robot* 20(7):849–866
57. Yang Z, Huo J, Monteiro H, Murray A (2009). Sensor-driven neuromorphic walking leg control. In: *IEEE international symposium on circuits and systems (ISCAS)*, pp 2137–2140
58. Yang Z, Cameron K, Lewinger W, Webb B, Murray AF (2012) Neuromorphic control of stepping pattern generation: a dynamic model with analog circuit implementation. *IEEE Trans Neural Net Learn Syst* 23(3):373–384
59. Twickel AV, Hild M, Siedel T, Patel V, Pasemann F (2012) Neural control of a modular multi-legged walking machine: simulation and hardware. *Robotics Auton Syst* 60(2):227–241
60. Barron-Zambrano JH, Torres-Huitzil C (2013) FPGA implementation of a configurable neuromorphic CPG-based locomotion controller. *Neural Netw* 45:50–61
61. Harris-Warrick RM (2011) Neuromodulation and flexibility in central pattern generator networks. *Curr Opin Neurobiol* 21:1–8
62. Bay JS, Hemami H (1987) Modeling of a neural pattern generator with coupled nonlinear oscillators. *IEEE Trans Biomed Eng* 34:297–306
63. Linkens DA, Taylor Y, Duthie HL (1976) Mathematical modeling of the colorectal myoelectrical activity in humans. *IEEE Trans Biomed Eng* 23:101–110
64. Tsutsumi K, Matsumoto H (1984) A synaptic modification algorithm in consideration of the generation of rhythmic oscillation in a ring neural network. *Biol Cybern* 50:419–430
65. Taga G, Yamaguchi Y, Shimizu H (1991) Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biol Cybern* 65:147–159
66. Taga G (1995) A model of the neuro-musculo-skeletal system for human locomotion—I. Emergence of basic gait. *Biol Cybern* 73:97–111
67. Amrollah E, Henaff P (2010) On the role of sensory feedbacks in rowat-selverston CPG to improve robot legged locomotion. *Front Neurobotics* 4:113. doi:[10.3389/fnbot.2010.00113](https://doi.org/10.3389/fnbot.2010.00113)
68. Schöner G, Jiang WY, Kelso JAS (1990) A synergetic theory of quadrupedal gaits and gait transitions. *J Theor Biol* 142:359–391
69. Collins JJ, Richmond SA (1994) Hard-wired central pattern generators for quadrupedal locomotion. *Biol Cybern* 71:375–385
70. Collins JJ, Stewart IN (1993) Coupled nonlinear oscillators and the symmetries of animal gaits. *J Nonlinear Sci* 3:349–392
71. Kimura H, Fukuoka Y, Cohen AH (2007) Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *Int J Robot Res* 26:475–490
72. Collins JJ (1995) Gait transitions. In: *Arbib MA (ed) The handbook of brain theory and neural networks*. The MIT Press, New York, pp 420–423
73. Yang Z, França FMG (1998) Generating arbitrary rhythmic patterns with purely inhibitory neural networks. In: *Proceedings of European symposium on artificial neural networks (ESANN)*, pp 53–58
74. França FMG, Yang Z (2000) Building artificial CPGs with asymmetric Hopfield networks. In: *Proceedings of international joint conference on neural networks, Como, Italy, vol IV*, pp 290–295
75. Yang Z, França FMG (2003) A generalized locomotion CPG architecture based on oscillatory building blocks. *Biol Cybern* 89(1):34–42
76. Barbosa VC, Gafni E (1989) Concurrency in heavily loaded neighbourhood-constrained systems. *ACM Trans Program Lang Syst* 11(4):562–584
77. Barbosa VC (1996) *An introduction to distributed algorithms*. The MIT Press, Cambridge
78. França FMG (1994) *Neural networks as neighbourhood-constrained systems*, unpublished doctoral dissertation. Imperial College, London
79. Barbosa VC, Benevides MRF, França FMG (2001) Sharing resources at nonuniform access rates. *Theory Comput Syst* 34(1):13–26
80. Kuffler SW, Eyzaguirre C (1955) Synaptic inhibition in an isolated nerve cell. *J Gen Physiol* 39:155–184
81. Perkel DH (1976) A computer program for simulating a network of interacting neurons: I. Organization and physiological assumptions. *Comput Biomed Res* 9:31–43
82. Roberts A, Tunstall MJ (1990) Mutual re-excitation with post-inhibitory rebound: a simulation study on the mechanisms for locomotor rhythm generation in the spinal cord of *Xenopus* embryos. *Eur J Neurosci* 2:11–23
83. Hopfield JJ (1982) Neural networks and physical systems with emergent collective properties. *Proc Natl Acad Sci USA* 79:2554–2558
84. Hopfield JJ, Tank DW (1986) Computing with neural circuits: a model. *Science* 233:625–632
85. Hopfield JJ, Tank DW (1985) Neural computation of decisions in optimization problems. *Biol Cybern* 52:141–152
86. Chen W, Ren G, Zhang J, Wang J (2012) Smooth transition between different gaits of a hexapod robot via a central pattern generators algorithm. *J Intell Robot Syst* 67:255–270
87. Getting PA (1989) Emerging principles governing the operation of neural networks. *Annu Rev Neurosci* 12:185–204
88. Soffe SR, Roberts A, Li WC (2009) Defining the excitatory neurons that drive the locomotor rhythm in a simple vertebrate: insights into the origin of reticulospinal control. *J Physiol* 587(20):4829–4844
89. Chua LO, Yang L (1988) Cellular neural networks: theory. *IEEE Trans Circuits Syst Part I* 35:1257–1272
90. Chua LO, Yang L (1988) Cellular neural networks: application. *IEEE Trans Circuits Syst Part I* 35:1273–1290
91. Pearson KG (1976) The control of walking. *Sci Am* 235:72–86
92. Patrick SK, Noah JA, Yang JF (2009) Interlimb coordination in human crawling reveals similarities in development and neural control with quadrupeds. *J Neurophysiol* 101(2):603–613
93. Ivanenko YP, Labini FS, Cappellini G et al (2011) Gait transition in simulated reduced gravity. *J Appl Physiol* 110:781–788
94. Hückesfeld S, Schoofs A, Schlegel P, Miroshnikov A, Pankratz MJ (2015) Localization of motor neurons and central pattern generators for motor patterns underlying feeding behavior in *Drosophila* Larvae. *PLoS One* 10(8):e0135011. doi:[10.1371/journal.pone.0135011](https://doi.org/10.1371/journal.pone.0135011)
95. Yu J, Tan M, Chen J, Zhang J (2014) A survey on CPG-Inspired control models and system implementation. *IEEE Trans Neural Netw Learn Syst* 25(3):441–456
96. Wang T, Guo W, Li M et al (2012) CPG control for biped hopping robot in unpredictable environment. *J Bionic Eng* 9(1):29–38
97. Liu C, Chen Q, Wang D (2011) CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots. *IEEE Trans Syst Man Cybern B Cybern* 41(3):867–880
98. Aoi S, Egi Y, Sugimoto R, Yamashita T et al (2012) Functional roles of phase resetting in the gait transition of a biped robot from quadrupedal to bipedal locomotion. *IEEE Trans Robot* 28(6):1244–1259

99. Ajallooeian M, Kieboom JVD, Mukovskiy A et al (2013) A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape. *Phys D* 263(15):41–56
100. Dzeladini F, Kieboom JVD, Ijspeert A (2014) The contribution of a central pattern generator in a reflex-based neuromuscular model. *Front Hum Neurosci*. doi:[10.3389/fnhum.2014.00371](https://doi.org/10.3389/fnhum.2014.00371)
101. Li F, Basu A, Chang C-H, Cohen AH (2012) Dynamical systems guided design and analysis of silicon oscillators for central pattern generators. *IEEE Trans Circuits Syst I Reg Pap* 59(12):3046–3059
102. Zhao L, Nogaret A (2014) Stimulus-dependent polyrhythms of central pattern generator hardware. *Int J Math Comput Phys Quantum Eng* 8(5):721–724