Journal of Intelligent & Robotic Systems (2019) 94:641–654
https://doi.org/10.1007/s10846-018-0864-y

Neural Oscillator Based CPG for Various Rhythmic Motions of Modular Snake Robot with Active Joints

Sajjad Manzoor[1,2] · Young Gil Cho[1,3] · Youngjin Choi[1]

Received: 17 May 2017 / Accepted: 30 April 2018 / Published online: 23 May 2018

Abstract
In this paper, construction of a newly designed snake robot is suggested along with the algorithm for generation of different rhythmic motions. The proposed robot system has modular structures with extendable length. It is subdivided into body, neck, head and tail modules. Each body module has two rotary motors to generate pitch and yaw motions of the snake robot. A linear actuator is also installed inside each body module in order to change the length of robot. The neck module in the robot is provided with two rotary motors to make the spherical motion of head module. Neural oscillator based central pattern generators (CPG) are used to produce rhythmic patterns for various snake robot movements, for example, serpentine, side-winding, two-step-concertina and four-step-concertina motions are generated in the snake robot using the proposed CPG algorithm. For serpentine motion, the body of robot is bent to form the planar sinusoidal waveform using whole body modules on the ground. To generate side-winding motion, the robot body is bent in such a way that it makes two dimensional sinusoidal waveform and only a few points of its body make contacts with the ground. By using the CPG algorithm, these contact points are propagated from tail to head, and the robot is ultimately moved along one side. In concertina motion, the body is sequentially pushed forward in steps from tail to head using the proposed algorithm. Finally several experiments are conducted on a laboratory floor in order to confirm the authenticity of robot design and CPG algorithm so that the comparison between different motions can be achieved.

Keywords Neural oscillator · Modular snake robot · Central pattern generator (CPG)

1 Introduction

Many snake robots have been designed and developed in the past few decades. This is due to the fact that snake belongs to animal category which has a flexible body without limbs, and has a very simple but a redundant body structure. These features help the snake in moving through narrow gaps, slithering on ground as well as swimming inside and on the surface of water. A snake can move with serpentine, concertina, rectilinear and side-winding types of motion [1]. Thus a robot that mimics a real snake and possesses the characteristics and features of an actual snake may be helpful in many real life applications such as inspections: inside the confined spaces, in pipelines, on the overhead power-lines, and rescue works: in unsafe areas, in man-made or natural disasters, furthermore, it can be used for surveillance purposes. The snake has a tendency to move on sand as well as on rocky mountains, and these properties make snake robots to be more appropriate as exploration robot.

Five types of snake robots have been thus far constructed as defined in [2, 3]. Most common snake robots belong to the class of robot with active joints as suggested in [4–7]. These kind of robots are much closer to the real snake in viewpoint of structure and function. Some snake robots have been also designed with active joints that

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10846-018-0864-y) contains supplementary material, which is available to authorized users.

✉ Youngjin Choi
cyj@hanyang.ac.kr

Sajjad Manzoor
sajjad.ee@must.edu.pk

Young Gil Cho
brian111001@gmail.com

[1] Department of Electronic Systems Engineering, Hanyang University, Ansan 15588, South Korea

[2] Faculty of Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, AJK, Pakistan

[3] Korea Institute of Science and Technology (KIST), Seoul 02792, South Korea

could actively bend and elongate the robot body[8]. The modular snake robot to be proposed and constructed in this paper possesses the properties of yaw-pitch rotations and longitudinal displacement, with the help of two rotary actuators and linear motor installed in each module [9]. Thanks to the linear motor, it is able to change the length of each body module which, ultimately, results in changing the length of snake robot.

Motions of real snake are rhythmic in nature. In animals, repetitive patterns for rhythmic motion generation are referred to as Central Pattern Generator (CPG). These CPG's could be generated mathematically and used to control the motion in bioinspired robots due to their simplicity, stability and easiness to generate different types of motions [10–13]. The CPG based serpentine motion generation has been achieved in snake robots in [14–19]. Recently some works have been conducted on the CPG based algorithm for side-winding motion generation [20]. In [21], the authors have developed the CPG based neural oscillator to generate the snake motions and used it to generate side-winding motion in [22]. In this paper, the algorithm is extended so as to contain the serpentine, side-winding, two-step-concertina and four-step-concertina motions. The extended algorithm is then applied to a newly constructed snake robot.

This paper is further organized as follows; Section 2 describes rhythmic serpentine motion, side-winding motion, two-step-concertina motion and four-step-concertina motion for snake robot, Section 3 suggests the design and construction of modular snake robot, Section 4 proposes the CPG based control algorithm for generation of the rhythmic motion in snake robot, Section 5 shows several experimental results about the motions of snake robot in order to confirm the effectiveness of the suggested algorithm and, finally, Section 6 concludes the paper.

2 Snake Locomotions

Snakes can move using different modes of locomotions, for example, they can swim in water as well as move on land with serpentine, side-winding, concertina, rectilinear types of motions [1]. In the paper, the rectilinear motion is ignored since it is very slow.

2.1 Serpentine Motion

Serpentine or lateral undulation motion is most common mode of movement in snakes. If it is seen from the top view, the body of snake makes a sinusoidal waveform as shown in Fig. 1a. The snake moves forward as if the sinusoidal waveform is sequentially propagated from tail to head along the body.

2.2 Side-Winding Motion

In side-winding motion, the snake forms two sinusoidal waveforms, the one is along the ground and the other is along the side of snake body, as shown in Fig. 1b. In this way, all the portions of snake body are not simultaneously in contact with the ground. Only two or three portions of its body are just contacted to the ground at the same time. Normally, the center portion of sinusoidal waveform touches the ground in the top-view as shown in the Fig. 1b. These ground contact portions are then propagated toward the head along the body.

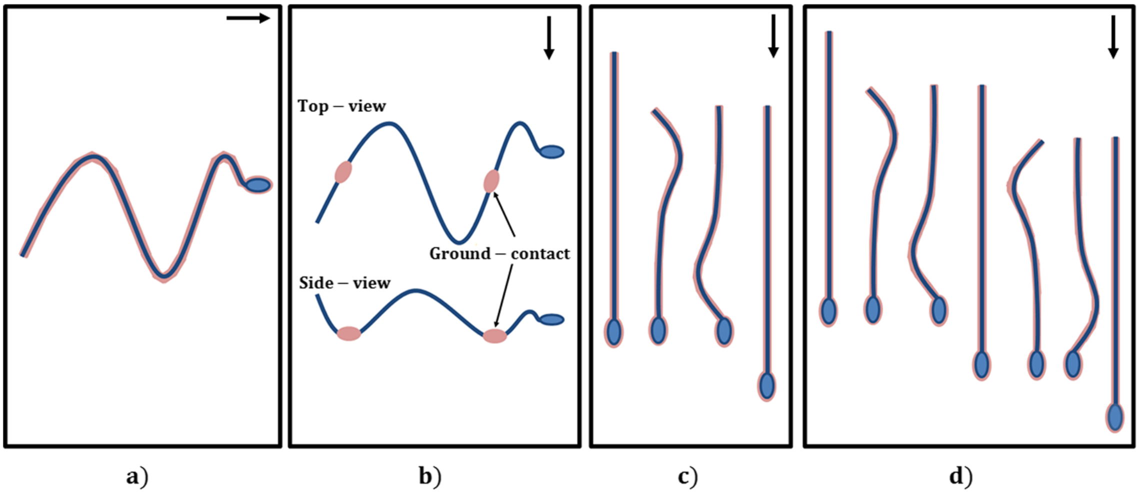

Fig. 1 Snake robot motions (arrow shows direction of motion of snake); **a** Serpentine motion (top-view), **b** side-winding motion (top-view and side-view), **c** two-step concertina motion (top-view), and **d** four step concertina motion (top-view)

The body of snake along with head and tail is used for side-winding motion generation. With the snake anchored at both head and tail, a body contact to ground is propagated from tail to head. The head and tail move alternatively back and forth during the motion. It allows the snake to move forward, in direction normal to the body, without slipping or sliding away.

2.3 Concertina Motions

Concertina is a special type of multistage snake propagation unlike continuous serpentine movement. Two-step and four-step concertina motions are considered in the paper.

Two-step Concertina Motion: During one full gait cycle, the snake moves forward in two steps of folding as shown in Fig. 1c. In the first step, the front-half portion of snake body acts as an anchor and the hind-half body part is folded. In the second step, the hind-half part of the body acts as an anchor while unfolding, and the front-half body is folded, and at the end it is unfolded to straighten the snake body. In this way, the body moves forward utilizing the friction increase between the snake body and ground during the anchoring phase. In this two-step concertina motion, the body of snake is bent in the same direction during each sequence of gait. This may result in side-wise motion of snake robot in place of the straight motion. In order to avoid the deviation from straight motion, two more steps are added into the gait cycle of concertina motion.

Four-step Concertina Motion: During the first two steps in the gait cycle, the snake robot is bent in one direction and then straightened. In the second set of steps, the snake robot is bent in other direction and then straightened as can be seen in Fig. 1d. In this way, the snake moves forward in a straight line.

3 Modular Snake Robot

The proposed snake robot consists of several modules and its final form is shown in Fig. 2 [9]. Snake robot is designed such that its length can be changed from 160.7 [cm] to 178.2 [cm], but the length is fixed to 160.7 [cm] during the experiments in this paper. The geometric parameters and kinematic limits of the robot are suggested in Table 1. The robot is constructed by using 3D printer and it can be subdivided into body, neck, tail and head modules.

3.1 Body Modules

A body-module is shown in Fig. 3a [9]. The body of snake robot is constructed by connecting seven same modules between tail-module and neck-module. These modules can be added or removed in order to change the length of robot.

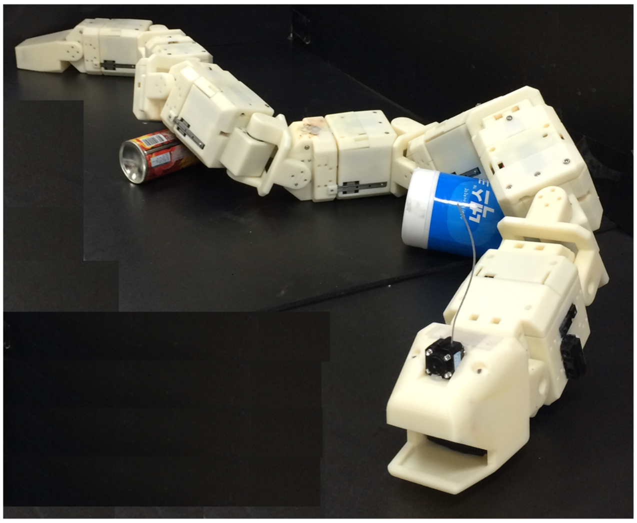

Fig. 2 Extendable 3D modular snake robot with active joints

The weight of body module is 0.68 [kg]. Each body-module has a width of 8 [cm], a height of 8 [cm] and the length ranged from 18.5 [cm] to 21 [cm]. Each module is equipped with a linear motor along with two linear guides. Each

Table 1 Geometric parameters and kinematic limits of modular snake robot, where it is noted that the length of each body-module is kept fixed to 18.5 [cm] and both head/tail modules denoted by (*) are tapered

Parameter	Value
Number of body-modules	7
Number of head-module	1
Number of neck-module	1
Number of tail-module	1
Width of each body-module (W_B)	8 [cm]
Width of neck-module (W_N)	8 [cm]
Maximum width of head-module* (W_H)	8 [cm]
Maximum width of tail-module* (W_T)	8 [cm]
Height of each body-module (H_B)	8 [cm]
Height of neck-module (H_N)	8 [cm]
Maximum height of head-module* (H_H)	8 [cm]
Maximum height of tail-module* (H_T)	6 [cm]
Length of every body-module (L_B)	18.5-21 [cm]
Length of head-module (L_H)	10 [cm]
Length of neck-module (L_N)	11.2 [cm]
Length of tail-module (L_T)	10 [cm]
Maximum allowable length of prismatic joint	2.5 [cm]
Weight of every body-module (W_B)	0.68 [kg]
Weight of neck-module (W_N)	0.5 [kg]
Weight of head-module (W_H)	0.22 [kg]
Weight of tail-module (W_T)	0.1 [kg]
Push/pull force of Linear Actuator	40 [N]
Range of yaw-joint angle φ_y	– 60 to 60 [deg]
Range of pitch-joint angle φ_p	-35 to 35 [deg]

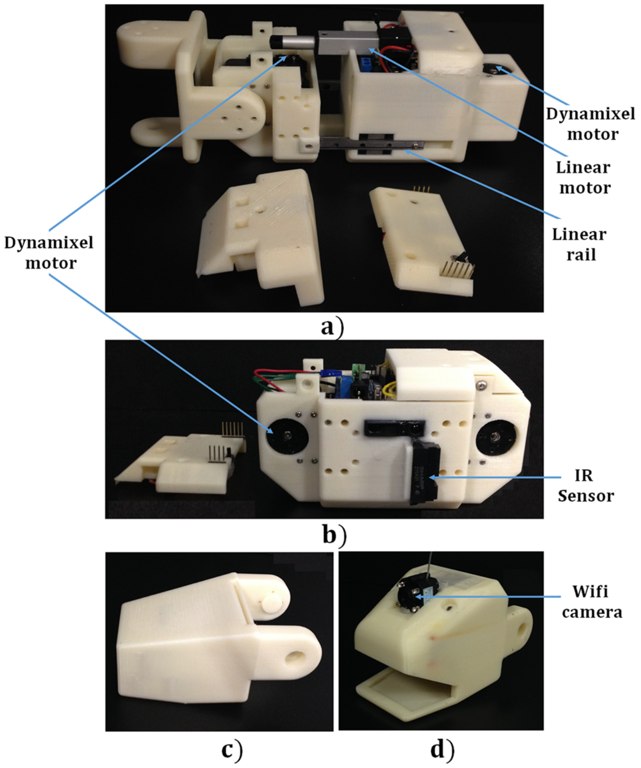

Fig. 3 Modules of the snake robot, where **a** body-module, **b** neck-module, **c** tail-module, and **d** head-module

linear actuator has maximal 3 [cm] stroke length and only 2.5 [cm] of stroke of linear actuator is utilized. Thus the length of each module and, ultimately, that of the robot can be changed by using linear actuator without adding or removing any module.

On the other hand, each linear actuator has push/pull force of 40 [N]. Each body-module also contains two rotary dynamixel motors (the one for yaw and the other for pitch rotation, respectively) each with a stall torque 1.8 [Nm]. A rechargeable battery and a microprocessor board is also included in the module. All body-modules have FSR (force-sensitive resistor) sensors at their bottom. Industrial adhesive tape is used to put on these FSR sensors. This tape is in direct contact with the ground and increases the contact-friction between the ground and the bottom surface of snake robot.

3.2 Neck Module

The snake robot has a single neck-module placed between head-module and front-most body-module as shown in Fig. 3b. This module has two dynamixel rotary motors, with stall torque of 1.8 [Nm], fixed in such a way that they provide only the pitch (or lift) motions of neck itself and head, respectively. Similar to the body-module, the neck-module has a height of 8 [cm] and a width of 8[cm], but its length is fixed to 11.2 [cm] because it does not have any linear actuator. This module also has a rechargeable battery and a microprocessor board. Along with this, two IR (infra-red) sensors facing in side-wise directions of robot body are provided on each side of the neck-module in order to avoid the collision of snake with walls or obstacle on each side. In addition two more IR sensors facing in front direction are installed to keep checking the obstacles in front.The weight of neck-module is 0.5 [kg]. This module also has the layers of FSR sensors and industrial adhesive tape at its bottom.

3.3 Tail and Head Modules

The robot has a tail-module and a head-module as shown in Fig. 3c and d, respectively. Both are tapered, in detail, the maximum height and width of head-module are all 8[cm], while the maximum height and width of tail-module are 6[cm] and 8[cm], respectively. The tail-module does not have any electrical components in itself, while the head-module is equipped with small wireless camera and 9 [V] battery. The tail-module might have a yaw motion with the help of actuator in the adjacent body-module, while the head-module is able to move in upward and downward direction with the help of pitch motor in the neck-module. The weights of tail- and head-module are 0.1 [kg] and 0.22 [kg], respectively. The industrial adhesive tapes are used at the bottom to increase the contact-friction between snake robot and ground.

3.4 Communication Between Modules

Different modules should communicate with each other in order to generate the coordinated motions in themselves. These modules in the snake robot communicate with each other using the I^2C communication protocol. The block digram for master-slave based configuration of I^2C communication is illustrated in Fig. 4. The microprocessor located in the neck-module plays a role of the master unit, while the microprocessors in body modules serve as the slave units. Desired orientations of all the motors are

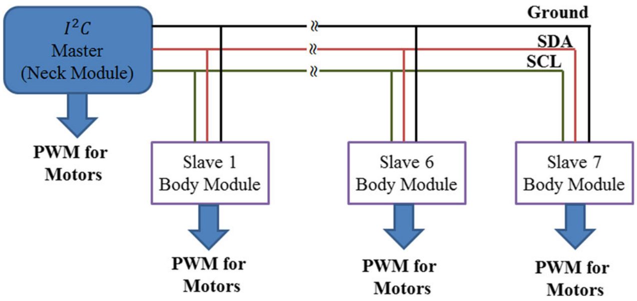

Fig. 4 I^2C protocol used for communication in-between modules of snake robot

calculated in the master unit using the CPG algorithm to be proposed in next section. These desired orientations of each yaw and pitch actuators are then transmitted to the slave units of body modules, in which they are converted into the PWM (pulse-width modulation) signals to operate the corresponding motors.

4 CPG Algorithm for Snake Robot Motions

Like other animals, a snake moves with a repetitive and rhythmic motion. The rhythmic serpentine, side-winding, two-step concertina and four-step concertina motions can be obtained in a snake robot by using the coupled excitatory and inhibitory neural oscillator based CPG. For a phase angle θ_{j_x} of j_xth neural oscillator, Kuramoto model in [23] with entrainment can be used as following form:

$$\dot{\theta}_{j_x} = \omega_{j_x} + \sum_{k=1}^{n} W_{j_x,k_x} \sin(\theta_{k_x} - \theta_{j_x} - \phi_{k_x,j_x}) \tag{1}$$

for $j_x = 1_x, 2_x, \cdots, n_x$, where n implies the total number of neural oscillators, θ_{k_x} is the phase angle of k_xth neural oscillator, ω_{j_x} is the natural frequency of j_xth neural oscillator, W_{j_x,k_x} denotes the coupling strength between j_xth and k_xth oscillators and ϕ_{k_x,j_x} represents the phase difference between k_xth and j_xth oscillators. Furthermore the subscript x can be replaced with y to represent for the yaw actuator, p the pitch actuator, N the neck actuator, and H the head actuator.

The natural frequency ω_{j_x} controls the speed of completion of each cycle and in this way it affects the speed of snake robot. By increasing the value of ω_{j_x}, the speed of completion of each gait cycle is increased. For synchronization for all neural oscillators, the value of ω_{j_x} is kept all the same. On the other hand, the coupling strength W_{j_x,k_x} gives an influence to the convergence rate as well as the synchronization rate of each neural oscillator.

4.1 Desired Orientations

Continuous firing neurons are required for the serpentine and side-winding motions, while the intermittent firing neurons with refractory period are needed for both two-steps and four-steps concertina motions. Aforementioned both types of neurons can be obtained by using a sigmoid activation function as following form:

$$f_{i_x}\left(\theta_{j_x}\right) = \frac{1}{1 + e^{(b_{j_x} - a_{j_x} \cos\theta_{j_x})}} \tag{2}$$

where $f_{i_x}(\theta_{j_x})$ has the value ranged between 0 and 1, the parameters a_{j_x} and b_{j_x} are helpful in generating the refractory period and determining the rise and fall [21] in output. Details for selection of parameters a_{j_x} and b_{j_x} are given in Appendix A.

The neural oscillator given in Eq. 1 can be used in the network as shown in Fig. 5. There are a pair of excitatory and inhibitory neural oscillators for each rotary actuator. It means that 32 neural oscillators are totally used in the network of snake robot. In detail, there are 14 coupled oscillators for seven yaw actuators, 14 for seven pitch actuators of body-modules, and 2 for each neck and head actuators in the neck-module, respectively. The neural oscillator network along with activation function of Eq. 2 generates the desired orientations for 16 actuators in the snake robot. In this way, a variety of locomotion types can be generated for the snake robot.

4.2 Desired Angles for Yaw and Pitch Motors

The desired orientation of i_yth actuator that generates the yaw motion, denoted by ϑ_{di_y}, is obtained from the difference between two coupled excitatory and inhibitory neurons of same actuator as follows:

$$\vartheta_{di_y} = \lambda_y(f_{(j+1)_y} - f_{j_y}) \tag{3}$$

where $j_y = 2i_y - 1$ for $i_y = 1_y, 2_y, \cdots, 7_y$ and the parameter λ_y adjusts the amplitude of ϑ_{di_y}. The desired orientation of i_pth actuator that generates the pitch motion, denoted by ϑ_{di_p}, is obtained from the difference between two coupled excitatory and inhibitory neuron of same actuator as follows:

$$\vartheta_{di_p} = \lambda_p(f_{(j+1)_p} - f_{j_p}) \tag{4}$$

where $j_p = 2i_p - 1$ for $i_p = 1_p, 2_p, \cdots, 7_p$ and the parameter λ_p adjusts the amplitude of ϑ_{i_p}. The desired orientation denoted by ϑ_{d_N} for neck actuator is given as:

$$\vartheta_{d_N} = \lambda_N(f_{2_N} - f_{1_N}) \tag{5}$$

and the desired orientation, ϑ_{d_H}, for head actuator is given by:

$$\vartheta_{d_H} = \lambda_H(f_{2_H} - f_{1_H}). \tag{6}$$

The desired orientations of all the actuators are calculated using Eqs. 3, 4, 5 and 6 inside the microprocessor of neck-module. Then they are transferred to body modules using I^2C communication protocol as illustrated in Fig. 4.

4.3 Phase-Difference ϕ_{k_x,j_x}

The neural oscillators in the proposed network, shown in Fig. 5, should have a phase difference ϕ_{k_x,j_x} in order to generate the specific type of motion in robot. Phase difference ϕ_{k_x,j_x} determines the divergence between two neuron firings, in time distribution at a scale of $0 \sim 2\pi$ in one gait cycle. For example, in cases of planar

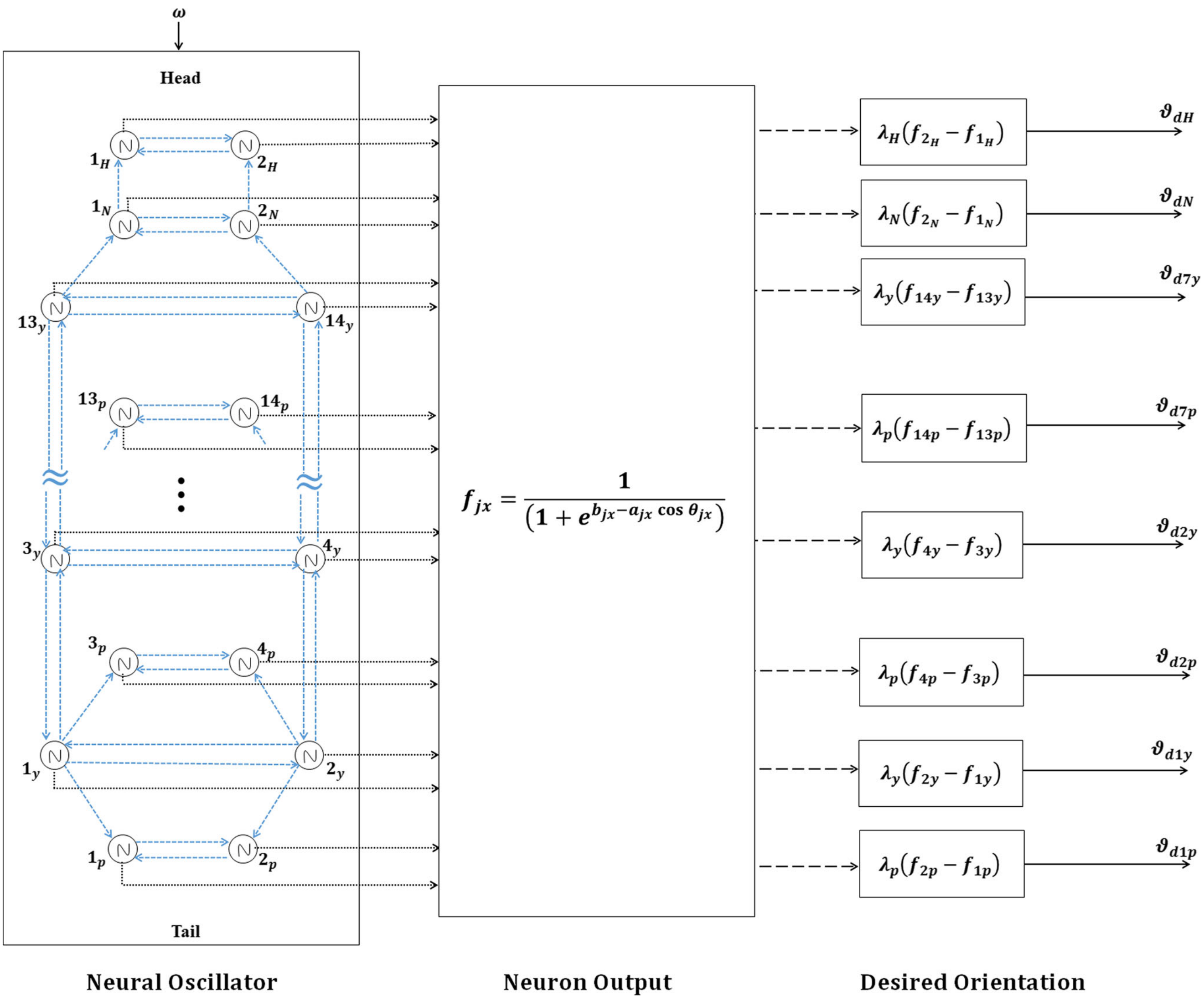
ω
Head
1_H
2_H
1_N
2_N
13_y
14_y
13_p
14_p
3_y
4_y
3_p
4_p
1_y
2_y
1_p
2_p
N
Tail
$f_{jx} = \frac{1}{(1 + e^{b_{jx} - a_{jx} \cos \theta_{jx}})}$
$\lambda_H(f_{2_H} - f_{1_H})$
$\lambda_N(f_{2_N} - f_{1_N})$
$\lambda_y(f_{14y} - f_{13y})$
$\lambda_p(f_{14p} - f_{13p})$
$\lambda_y(f_{4y} - f_{3y})$
$\lambda_p(f_{4p} - f_{3p})$
$\lambda_y(f_{2y} - f_{1y})$
$\lambda_p(f_{2p} - f_{1p})$
ϑ_{dH}
ϑ_{dN}
ϑ_{d7y}
ϑ_{d7p}
ϑ_{d2y}
ϑ_{d2p}
ϑ_{d1y}
ϑ_{d1p}
Neural Oscillator
Neuron Output
Desired Orientation

Fig. 5 Neural oscillator network for rhythmic motion generation in modular snake robot

serpentine, two-step and four-step concertina motions, the phase differences between the coupled excitatory and inhibitory neurons for each pitch rotation actuators in the network are taken such that $\phi_{k_p,j_p} = 0$, $\phi_{k_N,j_N} = 0$ and $\phi_{k_H,j_H} = 0$, which would result in constant zero orientations of these actuators. On the other hand, the non-planar side-winding motion makes use of $\phi_{k_p,j_p} = \phi_{k_N,j_N} = \phi_{k_H,j_H} = \pi$ so that the desired orientations for these actuators can be non-zero. Similarly, in order to form sine-wave shape in serpentine motion, the phase differences between the neighboring neurons in the network are taken by using the principle $\phi_{k_y,j_y} = 2\pi \frac{n_{sw}}{n_y}$ for the yaw actuators, where n_y is the total number of yaw actuators and n_{sw} is the total number of sinusoidal waveforms generated by snake robot body during the serpentine motion. More details about the selection criteria for values of phase difference ϕ_{k_x,j_x} are given in Appendix B, where it is noted that $\phi_{j_x,k_x} = -\phi_{k_x,k_x}$. In Table 2 the phase differences ϕ_{j_x,k_x} between j_xth and k_xth oscillators are given in order to generate serpentine, side-winding, two-step concertina and four-step concertina in the snake robot.

5 Results

The CPG based patterns and the desired orientations for different locomotions are illustrated in Fig. 6. These output profiles are generated in MatLab environment by using the proposed CPG algorithm. The phase differences ϕ_{k_x,j_x} given in the Table 2 were utilized. We take $a_{j_x} = 1$ and $b_{j_x} = 0$ for all the neurons in serpentine and side-winding motions. For the two-step concertina motion, $a_{j_p} = b_{j_p} = a_{j_N} = a_{j_H} = 3$ are set to the neurons of pitch actuators and $a_{j_y} = 3$, $b_{2_y} = b_{4_y} = b_{5_y} = b_{7_y} = b_{8_y} = b_{9_y} = b_{11_y} =$

Table 2 Phase difference ϕ_{k_x,j_x} between the neurons for serpentine, side-winding, two-step and four-step concertina locomotions, where it is noted that $\phi_{j_x,k_x} = -\phi_{k_x,k_x}$

j_x	k_x	ϕ_{k_x,j_x} Serpentine	ϕ_{k_x,j_x} Side-winding	ϕ_{k_x,j_x} Two-step concertina	ϕ_{k_x,j_x} Four-step concertina
1_p	2_p	0	π	0	0
1_y	2_y	π	π	0	π
1_y	1_p	0	$-\frac{3\pi}{14}$	0	0
2_y	2_p	$-\pi$	$-\frac{3\pi}{14}$	0	$-\pi$
1_y	3_p	0	$\frac{\pi}{2}$	0	0
2_y	4_p	π	$\frac{\pi}{2}$	0	π
3_p	4_p	0	π	0	0
1_y	3_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	0
2_y	4_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	0
3_y	4_y	π	π	0	π
3_y	5_p	0	$\frac{\pi}{2}$	0	0
4_y	6_p	π	$\frac{\pi}{2}$	0	π
5_p	6_p	0	π	0	0
3_y	5_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	π
4_y	6_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	π
5_y	6_y	π	π	0	π
5_y	7_p	0	$\frac{\pi}{2}$	0	0
6_y	8_p	π	$\frac{\pi}{2}$	0	π
7_p	8_p	0	π	0	0
5_y	7_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	0
6_y	8_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	π
7_y	8_y	π	π	0	0
7_y	9_p	0	$\frac{\pi}{2}$	0	0
8_y	10_p	π	$\frac{\pi}{2}$	0	0
9_p	10_p	0	π	0	0
7_y	9_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	$\frac{\pi}{2}$	$\frac{3\pi}{2}$
8_y	10_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$
9_y	10_y	π	π	0	π
9_y	11_p	0	$\frac{\pi}{2}$	0	0
10_y	12_p	π	$\frac{\pi}{2}$	0	π
11_p	12_p	0	π	0	0
9_y	11_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	0
10_y	12_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	0
11_y	12_y	π	π	0	π
11_y	13_p	0	$\frac{\pi}{2}$	0	0
12_y	14_p	π	$\frac{\pi}{2}$	0	π
13_p	14_p	0	π	0	0
11_y	13_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	π
12_y	14_y	$\frac{2\pi}{7}$	$\frac{2\pi}{7}$	0	π
14_y	13_y	π	π	0	π
13_y	1_N	0	$\frac{\pi}{4}$	0	0
14_y	2_N	π	$\frac{\pi}{4}$	0	π
1_N	2_N	0	π	0	0
1_N	1_H	0	$\frac{\pi}{4}$	0	0
2_N	2_H	0	$\frac{\pi}{4}$	0	0
1_H	2_H	0	π	0	0

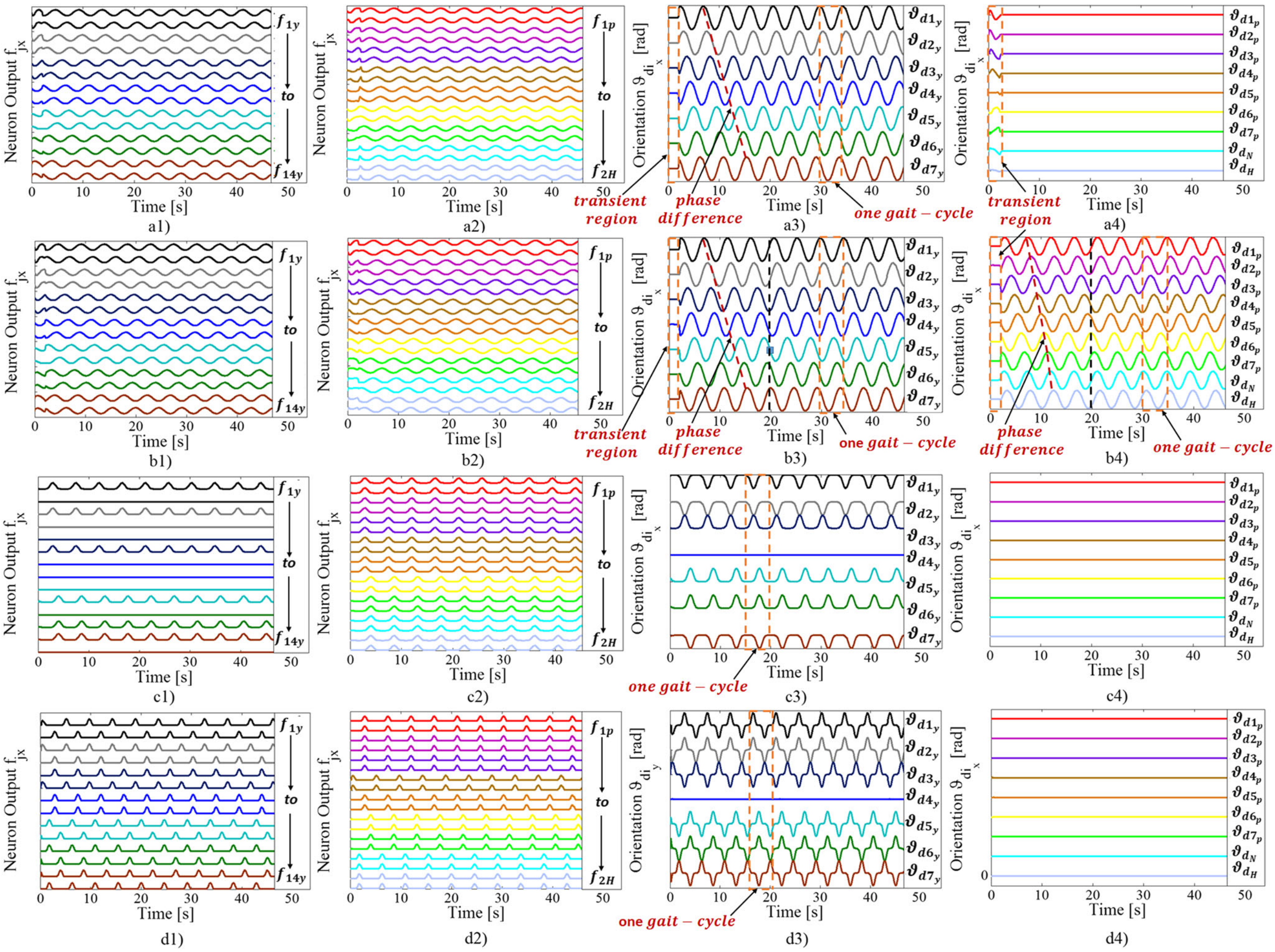

Fig. 6 Neural oscillator outputs and desired orientations; **a** serpentine motion; a1) neural oscillator outputs for yaw actuators, a2) neural oscillator outputs for pitch actuators, a3) desired orientations for yaw actuators, and a4) desired orientations for pitch actuators, **b** side-winding motion; b1) neural oscillator outputs for yaw actuators, b2) neural oscillator outputs for pitch actuators, b3) desired orientations for yaw actuators, and b4) desired orientations for pitch actuators, **c** two-step concertina motion; c1) neural oscillator outputs for yaw actuators, c2) neural oscillator outputs for pitch actuators, c3) desired orientations for yaw actuators, and c4) desired orientations for pitch actuators, and **d** four-step concertina motion; d1) neural oscillator outputs for yaw actuators, d2) neural oscillator outputs for pitch actuators, d3) desired orientations for yaw actuators, and d4) desired orientations for pitch actuators

$b_{14_y} = 15$ and $b_{1_y} = b_{3_y} = b_{6_y} = b_{10_y} = b_{12_y} = b_{13_y} = 3$ are set to the neurons of yaw actuators. On the other hand, $a_{j_x} = b_{j_x} = 8$ are set to obtain four-step concertina motion. The outputs of neural oscillators (generated from Eq. 2) and the desired orientations (generated from Eqs. 3, 4, 5 and 6) for cases of serpentine, side-winding, two-step and four-step concertina motions are illustrated in the Fig. 6.

It can be seen in the Fig. 6 that the desired orientations of yaw actuators for serpentine motion and side-winding locomotions have the sinusoidal waveforms, although the desired orientations of yaw actuators for two-step concertina motions have the refractory periods in them. In the first step, yaw actuators 1_y, 2_y and 3_y (1_y and 2_y in negative direction and 3_y in positive direction) are bent simultaneously and the orientations of actuators 5_y, 6_y and 7_y remain 0[rad]. Then, in the second step, yaw actuators 5_y, 6_y and 7_y (5_y and 6_y in positive direction and 7_y in negative direction) are bent simultaneously and the orientations of actuators 1_y, 2_y and 3_y remain 0[rad]. It is noted that the orientation of actuator 4_y is always 0[rad] to make the motion to be balanced. In the four-step concertina motion, a group of yaw actuators 1_y, 2_y and 3_y and another group of actuators 5_y, 6_y and 7_y are bent alternatively, once in one direction and secondly in other direction (twice in one gait-cycle). Here, the orientation of actuator 4_y remains also 0 [rad] for the motion balance.

On the other hand, the pitch actuators for serpentine and concertina motions have continuous zero orientations, which means that these are in planar motion. For side-winding motion, the orientations of pitch actuators including neck and head actuators are sinusoidal and it can

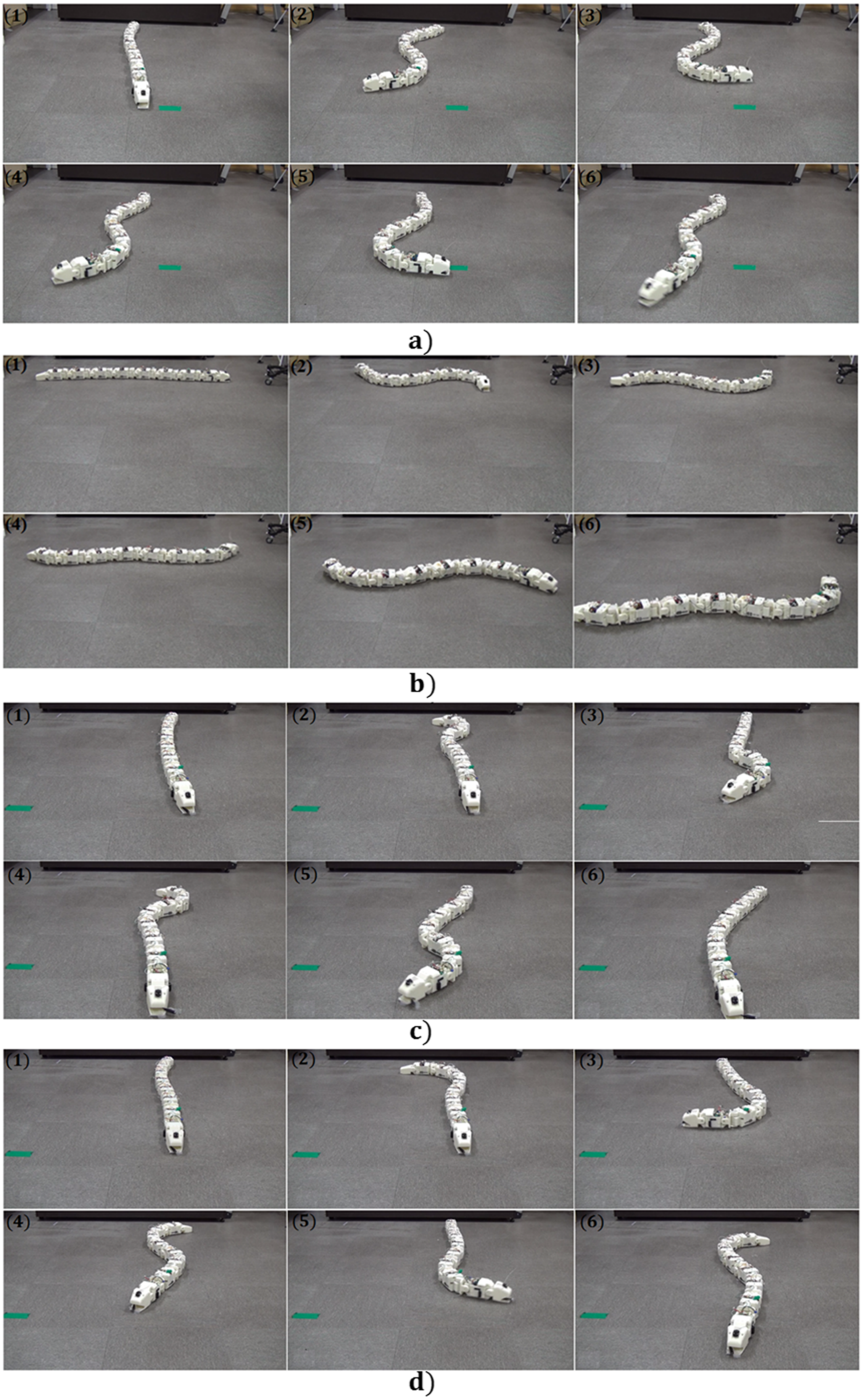

Fig. 7 Rhythmic motions of modular snake robot; **a** serpentine motion, **b** side-winding motion, **c** two-step concertina motion, and **d** four-step concertina motion

be seen that the ground contact of body is sequentially propagated from tail to head. Along with this, it can be seen that the yaw actuator 5_y is 0[rad] at any time instant, while the pitch actuator 6_p (closest to yaw actuator 5_y) with peak value is in contact with the ground. In this way, the conditions for side-winding motion are achieved.

5.1 Experimental Results

Experiments regarding the serpentine, side-winding, two-step and four-step concertina motions were conducted by using the modular snake robot of the Fig. 2. The desired orientations of yaw and pitch actuators for these locomotions were obtained by coding the proposed neural oscillator based CPG algorithm in the microprocessor located inside the neck-module. The communications between neck-module and body-modules were achieved by using I^2C protocol as suggested in the Section 2. Notice that the length of each linear motor was kept to be constant 0 [cm]. The phase differences between the neurons were set as suggested in the Table 2. The parameters $\lambda_y = 1$ and $\lambda_p = \lambda_N = \lambda_H = 0.5$ were determined. The natural frequencies for all the oscillators were taken such that $\omega_{j_x} = \frac{\pi}{2}[rad/s]$. In all the experiments, the robot was started from stationary position. The surface used for experiments was the smooth floor of the laboratory.

Figure 7a shows the sequential snapshots of serpentine motion generated in the snake robot by using the proposed algorithm. The snake robot moves forward making the sinusoidal waveform. Whole body of the snake robot always touches the ground, which means that it is in planar motion. The overall movement of robot is along the head direction of snake robot. From the experimental result conducted on smooth laboratory floor, it could be deduced that the robot moved in a rhythmic sinusoidal waveform but the forward motion was slow due to floor slippage and the lesser distance was covered by snake robot.

Figure 7b suggests the sequential snapshots of side-winding motion. It can be seen that the orientations of yaw actuators generated by the algorithm make the snake to be bent for the sinusoidal waveform, while the pitch actuators are controlled using the neural oscillators so that the ground contact of the robot can be propagated from tail to head. The final movement of robot is along one side of the snake body.

Figure 7c illustrates the two-step concertina motion obtained by using the proposed algorithm. It can be seen that the snake completes one gait cycle in two folding steps. In the first step, the hind half body of robot is bent, while the front half body acts like an anchor as shown in the Fig. 7c(2). Then the front half body of robot is bent when the hind half body is unfolded itself as shown in the Fig. 7c(3). At the end of gait cycle, the body of snake robot is straighten again by being the front half unfolded. The robot moves forward by repeating these kind of motions, but the snake body keeps being tilted in one direction, not moving in a straight line.

In Fig. 7d, the sequential snapshots of four-step concertina motion are suggested. In this type of concertina locomotion, one gait cycle consists of four steps. During the first step, the hind half of robot is bent in one way, while the front half acts like anchor as shown in the Fig. 7d(2). Then the front half is bent when the hind half is unfolded as shown in Fig. 7d(3). In the third step shown in the Fig. 7d(4), the hind half of robot is bent in the opposite direction with unbent front half body working as an anchor. During the fourth step suggested in the Fig. 7d(5), the front half is bent,

Fig. 8 Box-plot for time taken by snake robot to cover the same distance of 1.5[m] by using four types of motions

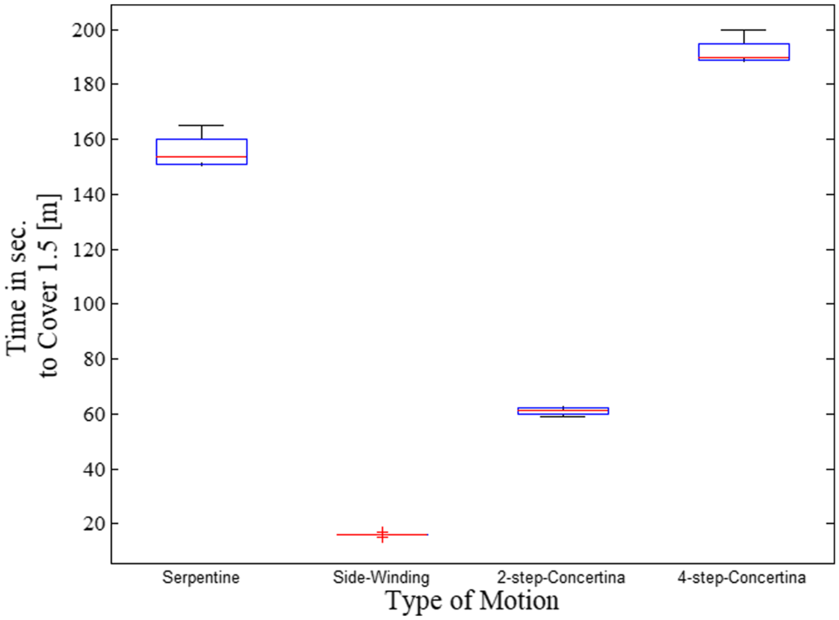

Table 3 Comparisons between existing previous experimental works on CPG based snake robot motion generation and the proposed algorithm

	Serpentine motion	Side-winding motion	Two-step concertina motion	Four-step concertina motion
Ijspeert A J et al. [14]	Yes	No	No	No
Ma S et al. [15–17]	Yes	No	No	No
Bing Z et al. [18]	Yes	No	No	No
Zhang D et al. [19]	Yes	No	No	No
Wang Z et al. [20]	Yes	Yes	No	No
The proposed	Yes	Yes	Yes	Yes

while the hind half of robot body is unfolded itself. In this way, the robot moves forward making the straight motion. In our experiments, we could observe the property regarding the locomotions. In each type of motion, the snake robot moves forward with different speeds as shown in Fig. 8. This figure shows box-plot for time taken by snake robot to cover same distance by using four types of motions. Both serpentine and four-step concertina motions were relatively slow in the low friction environments because the slippery ground does not make forward movement of robot well, but the side-winding and two-step concertina motions were relatively fast because they make point contacts or anchoring with the slippery ground irrespective of the friction coefficient.

6 Conclusions

In this paper a simple CPG algorithm was proposed for the generation of various types of rhythmic motions. Along with this, the construction of snake robot, that is able to move in three dimensional space, was suggested, including yaw and pitch actuators. The constructed snake robot was the modular type comprising of a head, a neck, a tail and seven body-modules. The unified CPG based algorithm for the serpentine, side-winding, two-step and four-step concertina motions was applied to the modular snake robot. Through experiments, we could confirm that the natural frequency of the serpentine motions could be controlled by adjusting the parameters. During the application of side-winding motion, the snake moved quicker in a direction perpendicular to the robot body. In two-step concertina motion, either hind or front body was always in anchoring state, thus we have bigger forward motion due to the friction increase, although the movement of snake was not guaranteed to be in straight line. As an alternative, the four-step concertina motion could be utilized for the straight line motion.

The comparisons of the proposed method with already existing CPG based snake robot motions are given in Table 3. It can be seen that, in all the previous experimental works, only one or two types of snake motion were considered. On the other hand, in this paper, CPG based serpentine, side-winding, two-step and four-step concertina motions were generated. Although some of the resulting different kind of motions were fast and some were slow, each motion had its own significance. By combining all the proposed motions in near future, the snake robot will be utilized for real applications which cannot be achieved by a single motion. Furthermore, the effect of friction and self-propelled motion generation and control of friction-induced stick-slip motion [24, 25] in the snake robot using the proposed algorithm would also be studied in our future work.

Acknowledgments This work was supported by the Convergence Technology Development Program for Bionic Arm through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Planning (NRF- 2015M3C1B2052811).

Appendix A: Selection of Parameters a_{j_x} and b_{j_x}

Neuron activation model of Eq. 2 is derived from the generalized sigmoid function given as:

$$A = \frac{1}{1+e^{-B}} \quad \begin{cases} \approx 0 & \forall\ B < -6 \\ = 0.5 & \forall\ B = 0 \\ \approx 1 & \forall\ B > 6 \end{cases}$$

where $A = f_{i_x}(\theta_{j_x})$ and $B = -(b_{j_x} - a_{j_x}\cos(\theta_{j_x}))$ as given in Eq. 2. In above equation, the parameters a_{j_x} and b_{j_x} determine the shape of output generated by neural activation. On the requirements of different types of snake robot motions, the values of parameters a_{j_x} and b_{j_x} are selected as follows [21]:

1. In serpentine and side-winding motions, $f_{i_x}(\theta_{j_x})$ is required to be a periodic function having range $0 < f_{i_x}(\theta_{j_x}) < 1$. This periodic function can be achieved by taking $b_{j_x} = 0$ and $-6 < a_{j_x} < 6$ for given $-1 \leq \cos(\theta_{j_x}) \leq 1$.
2. For two-step and four-step concertina motions, the neuron having refractory period is required due to discontinues motion of snake. If the values of

parameters are taken such that $a_{j_x} = b_{j_x} > 6$, then the function $f_{i_x}(\theta_{j_x})$ yields its output as follows:

$$f_{i_x}(\theta_{j_x}) \approx 0 \qquad \text{for } \cos\theta_{j_x} < \frac{a_{j_x} - 6}{a_{j_x}}$$

$$f_{i_x}(\theta_{j_x}) = 0.5 \qquad \text{when } \theta_{j_x} = 0$$

Appendix B: Selection of Phase Difference ϕ_{k_x, j_x}

1. Serpentine Motion:

(a) The serpentine motion is performed in the plane, i.e., pitch actuators as well as head and neck actuators have constant zero orientations. These can be achieved by keeping the coupled excitatory and inhibitory neurons for pitch actuators to be the same phases, namely, the phase differences ϕ_{k_p, j_p}, ϕ_{k_N, j_N} and ϕ_{k_H, j_H} for these actuators are taken such that:

$$\phi_{k_p, j_p} = \phi_{k_N, j_N} = \phi_{k_H, j_H} = 0 \quad \forall\ k = j + 1$$

(b) In case of yaw actuators, the coupled excitatory and inhibitory neurons are taken out of phase, in this way, they do not have any constant zero orientations and thus the snake forms a sine-wave shape on ground. The phase differences ϕ_{k_y, j_y} for yaw actuators are taken such that:

$$\phi_{k_y, j_y} = \pi \qquad \forall\ k = j + 1$$

On the other hand, the phase differences between the neighboring neurons in the network are taken by using this principle:

$$\phi_{k_y, j_y} = 2\pi \frac{n_{sw}}{n_y} \qquad \forall\ k = j + 2$$

with n_y as the total number of yaw actuators and n_{sw} as the total number of sinusoidal waveforms required during serpentine motion, namely, $n_y = 7$ and $n_{sw} = 1$ for our experiments.

(c) As the coupled neurons of pitch actuators and yaw actuators, since those of yaw actuators are out of phase and those of pitch actuators are in phase, the phase differences between odd pair of neurons of yaw actuator and neighboring pitch actuator are taken such that:

$$\phi_{k_p, j_y} = 0 \qquad \forall \text{ odd } k = j + 2$$

While those between even pair of neurons of yaw actuator and neighboring pitch actuator are taken such that:

$$\phi_{k_p, j_y} = \pi \qquad \forall \text{ even } k = j + 2$$

Similar conditions are applied to the phase differences between neurons of seventh yaw actuator and neck actuator as follows:

$$\phi_{1_N, 13_y} = 0, \quad \phi_{2_N, 14_y} = \pi$$

(d) On the other hand, the phase differences between neurons of first yaw actuator and first pitch actuator are taken such that:

$$\phi_{1_p, 1_y} = 0, \quad \phi_{2_p, 2_y} = -\pi$$

(e) The phase differences between neurons of neck actuator and the head actuator are taken such that:

$$\phi_{1_H, 1_N} = \phi_{2_H, 2_N} = 0$$

2. Side-winding Motion:

(a) The side-winding is a non-planer motion. This can be achieved by taking out of the phase difference between the coupled excitatory and inhibitory neurons of both pitch and yaw actuators. The phase differences between these neurons are taken such that:

$$\phi_{k_p, j_p} = \phi_{k_y, j_y} = \pi \qquad \forall\ k = j + 1$$

(b) In case of yaw actuators, the phase differences between the neighboring neurons in the network are taken by using this principle:

$$\phi_{k_y, j_y} = 2\pi \frac{n_{sw}}{n_y} \qquad \forall\ k = j + 2$$

where $n_y = 7$ and $n_{sw} = 1$ so that one full sine-wave can be generated by snake body during side-winding motion.

(c) The phase differences between neurons of yaw actuator and neighboring pitch actuator are taken such that:

$$\phi_{k_p, j_y} = \frac{\pi}{2} \qquad \forall\ k = j + 2$$

It results in the ground contact at the center of sine-wave form.

(d) The phase differences between neurons of first yaw actuator and first pitch actuator are taken such that:

$$\phi_{1_p, 1_y} = \phi_{2_p, 2_y} = -\frac{3\pi}{14}$$

(e) The phase differences between neurons of seventh yaw rotation actuator and neck actuator are taken such that:

$$\phi_{1_N, 13_y} = \phi_{2_N, 14_y} = \frac{\pi}{4}$$

While the phase differences between neurons of neck and head actuator are taken such that:

$$\phi_{1_H, 1_N} = \phi_{2_H, 2_N} = \frac{\pi}{4}$$

These result in a total of $\frac{\pi}{2}$ phase difference between foremost yaw actuator and two consecutive neck and head pitch actuators.

3. Two-Step Concertina Motion:

(a) Like serpentine motion, two-step concertina is planner motion, i.e., pitch actuators as well as head and neck actuators have constant zero orientations. Thus the phase differences for these actuators are taken such that:

$$\phi_{k_p,j_p} = \phi_{k_N,j_N} = \phi_{k_H,j_H} = 0 \qquad \forall\ k = j+1$$

(b) In case of yaw actuators, phase differences between the coupled excitatory and inhibitory neurons are taken such that:

$$\phi_{k_y,j_y} = 0 \qquad \forall\ k = j+1$$

(c) The phase differences between the neighboring neurons are taken to be $\phi_{k_y,j_y} = 0$ for actuators that fold during the same step, but $\phi_{9_y,7_y} = \phi_{10_y,8_y} = \frac{\pi}{2}$ for neighboring actuators that separate the front and hind folding parts of robot body.

4. Four-Step Concertina Motion:

(a) Like serpentine and two-step concertina motions, four-step concertina is also planner motion, i.e., pitch actuators as well as head and neck actuators have constant zero orientations. Thus the phase differences for these actuators are taken as:

$$\phi_{k_p,j_p} = \phi_{k_N,j_N} = \phi_{k_H,j_H} = 0 \qquad \forall\ k = j+1$$

(b) In case of yaw actuators, the coupled excitatory and inhibitory neurons are out of phase, in this way, they do not have any constant zero orientations. The phase differences for yaw actuators are taken such that:

$$\phi_{k_y,j_y} = \pi \qquad \forall\ k = j+1$$

except $\phi_{8_y,7_y} = 0$ for the phase difference between neurons located at the mid of yaw actuators.

(c) The phase differences between the neighboring neurons of yaw actuators are taken such that:

$$\phi_{k_y,j_y} = 0 \qquad \forall\ k = j+2$$

except $\phi_{9_y,7_y} = \frac{3\pi}{2}$ and $\phi_{10_y,8_y} = \frac{\pi}{2}$ for the phase differences between neighboring neurons located at the mid of yaw actuators.

(d) The phase differences between even pair of neurons of yaw actuator and neighboring pitch actuator are given as:

$$\phi_{k_p,j_y} = \pi \qquad \forall\ \text{even} \quad k = j+2$$

While the phase differences between odd pair of neurons of yaw actuator and neighboring pitch actuator are given as

$$\phi_{k_p,j_y} = 0 \qquad \forall\ \text{odd} \quad k = j+2$$

Similar conditions are applied to the phase differences between neurons of seventh yaw actuator and neck actuator as follows:

$$\phi_{1_N,13_y} = 0, \quad \phi_{2_N,14_y} = \pi$$

(e) On the other hand, the phase differences between neurons of first yaw actuator and first pitch actuator are taken such that:

$$\phi_{1_p,1_y} = 0 \quad \phi_{2_p,2_y} = -\pi$$

(f) The phase differences between neurons of neck and head actuators are taken such that:

$$\phi_{1_H,1_N} = \phi_{2_H,2_N} = 0$$

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Lilly white, H.B.: How Snakes Work: Structure, Function and Behavior of the World's Snakes. Oxford University Press, London (2014)
2. Hirose, S., Yamada, H.: Snake-like robots machine design of biologically inspired robots. IEEE Robot. Autom. Mag. **16**(1), 88–98 (2009)
3. Hopkins, J.K., et al.: A survey of snake inspired robot designs. Bioinspir. Biomim. **4**(2), 021001 (2009)
4. Crespi, A., et al.: Amphibot I: an amphibious snake-like robot. Mechatronics **50**(4), 163–175 (2005)
5. Crespi, A., Ijspeert, A.J.: Amphibot II: an amphibious snake robot that crawls and swims using a central pattern generator. In: International Conference on Climbing and Walking Robots, pp. 19–27 (2006)
6. Wright, C., et al.: Design and architecture of the unified modular snake robot. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4347–4354 (2012)
7. Pal, L., et al.: Mamba - a waterproof snake robot with tactile sensing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 294–301 (2014)
8. Sugita, S., et al.: A study on the mechanism and locomotion strategy for new snake-like robot active cord mechanism Slime model 1 ACM-s1. J. Rob. Mechatronics **20**(2), 302–310 (2008)
9. Manzoor, S., Choi, Y.: Modular design of snake robot for serpentine and rectilinear motions. In: 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 211–213 (2016)
10. Junzhi, Yu., et al.: A survey on CPG-inspired control models and system implementation. IEEE Transactions on Neural Networks and Learning Systems **25**(3), 441–456 (2014)
11. Rostro-Gonzalez, H., Cerna-Garcia, P.A., Trejo-Caballero, G., Garcia-Capulin, C.H., Ibarra-Manzano, M.A., Avina-Cervantes, J.G., Torres-Huitzil, C.: A CPG system based on spiking neurons for hexapod robot locomotion. Neurocomputing **170**, 47–54 (2015)

12. Yu, H., Gao, H., Ding, L., Li, M., Deng, Z., Liu, G.: Gait generation with smooth transition using CPG-based locomotion control for hexapod walking robot. IEEE Trans. Ind. Electron. **63**(9), 5488–5500 (2016)
13. Zhong, B., Zhang, S., Xu, M., Zhou, Y., Fang, T., Li, W.: On a CPG-based hexapod robot: AmphiHex-II with variable stiffness legs. IEEE/ASME Trans. Mechatron. **23**(2), 542–551 (2018)
14. Crespi, A., Ijspeert, A.J.: Online optimization of swimming and crawling in an amphibious snake robot. IEEE Trans. Robot. **24**(1), 75–87 (2008)
15. Wu, X., Ma, S., et al.: Adaptive creeping locomotion of a CPG-controlled snake-like robot to environment change. Auton. Robot. **28**(3), 283–294 (2010)
16. Wu, X., Ma, S.: CPG-based control of serpentine locomotion of a snake-like robot. Mechatronics **20**(2), 326–334 (2010)
17. Nor, N.M., Ma, S.: A simplified CPGs network with phase oscillator model for locomotion control of a snake-like robot. J. Intell. Robot. Syst. **75**(1), 71–86 (2014)
18. Bing, Z. et al.: Towards autonomous locomotion: CPG-based control of smooth 3D slithering gait transition of a snake-like robot. Bioinspir. Biomim. **12**(3), 035001 (2017)
19. Zhang, D. et al.: Smooth transition of the CPG-based controller for snake-like robots. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2716–2721 (2017)
20. Wang, Z. et al.: CPG-inspired locomotion control for a snake robot basing on nonlinear oscillator. J. Intell. Robot. Syst. **85**(2), 209–227 (2017)
21. Manzoor, S., Choi, Y.: A unified neural oscillator model for various rhythmic locomotions of snake-like robot. Neurocomputing **173**(3), 1112–1123 (2016)
22. Manzoor, S., Choi, Y.: Recurring side-winding motion generation for modular snake robot. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1468–1473 (2017)
23. Sksguchi, H., Kuramoto, Y.: A soluble active rotator model showing phase transitions via mutual entrainment. Lect. Notes Phys. **76**(3), 576–581 (1986)
24. Liu, P., Yu, H., Cang, S.: Modelling and dynamic analysis of underactuated capsule systems with friction-induced hysteresis. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 549–554 (2016)
25. Liu, P., Yu, H., Cang, S.: Geometric analysis-based trajectory planning and control for underactuated capsule systems with viscoelastic property. Trans. Inst. Meas. Control. **40**(7), 2416–2427 (2018)

Sajjad Manzoor received his BSc degree in Electrical Engineering from the University of Engineering and Technology, Lahore, Pakistan in 2007, and the MS degree in Electronics, Electrical, Control and Instrumentation Engineering and PhD degree in Electronic Systems Engineering from Hanyang University, Seoul, South Korea, in 2010 and 2016, respectively. Currently, he is an assistant professor at the department of Electrical (power) Engineering, Mirpur University of Science and Technology (MUST), Mirpur AJK, Pakistan. His research interest include biorobotics, multi-agent coordination, control and automation.

Young Gil Cho received his BS degree in Electronic Systems Engineering from Hanyang University, Ansan, South Korea, in 2016, and the MS degree in Electronic Systems Engineering from Hanyang University, Seoul, in 2018. Currently, he is a researcher at the Convergence Research Center for Diagnosis, Treatment and Care System of Dementia of the Korea Institute of Science and Technology (KIST). His research interest include biorobotics, machine learning and motion planning.

Youngjin Choi received his BS degree in the Precision Mechanical Engineering from Hanyang University, Seoul, Korea, in 1994, and MS and PhD degrees in the Mechanical Engineering from POSTECH, Pohang, Korea, in 1996 and 2002, respectively. Since 2005, he is a professor at the department of Electronic Systems Engineering of Hanyang University, Ansan, Korea. From 2002 to 2005, he was a senior research scientist at the intelligent robotics research center of the Korea Institute of Science and Technology (KIST). From 2011 to 2012, he was a visiting researcher at the University of Central Florida, USA. From 2010 to 2014, he was an associate editor of the IEEE Transactions on Robotics. His research interests include biorobotics, control theory, rehabilitation robot, and dual-arm manipulation.