Iranian Journal of Science and Technology, Transactions of Electrical Engineering (2020) 44:1093–1103
https://doi.org/10.1007/s40998-019-00301-8

RESEARCH PAPER

Serpentine and Rectilinear Motion Generation in Snake Robot Using Central Pattern Generator with Gait Transition

Sajjad Manzoor[1] · Uzair Khan[2] · Ihsan Ullah[2]

Received: 19 April 2019 / Accepted: 6 December 2019 / Published online: 19 December 2019

Abstract
This paper contributes to the fabrication of a snake-like robot in which the motion can be achieved through active wheels. The robot is constructed in such a way that its size can be increased and decreased, as well as it can undulate into a sine wave-like shape. The snake robot with wheels consists of chain of links attached to each other with the help of the passive prismatic and revolute joints. A neural oscillator-based central pattern generator (CPG) algorithm is applied to the robot so that rhythmic serpentine as well as rectilinear motions can be generated in it. In addition, a novel smooth transition mechanics of robot motion, from stationary position to serpentine or rectilinear motion as well as transition between these two gaits, is also suggested in the paper. The working of the formulated CPG motion algorithm is realized through experimental setup equipped with a motion capture system as well as through simulations.

Keywords Central Pattern Generator (CPG) · Snake robot · Neural oscilator · Gait transition

1 Introduction

Recently, many researches have been done on construction of snake robots and generation of algorithms for their locomotions (Javaheri et al. 2018). The snake is a popular inspiration for researchers since it has flexible body with number of joints in it. This property, along with it being limbless, gives the snake advantage to move in variety of environments. Furthermore, a snake can move with many different types of motions on ground such as the serpentine, side-winding, rectilinear, concertina locomotion, and it can swim as well as it can also jump.

✉ Sajjad Manzoor
sajjad.ee@must.edu.pk

Uzair Khan
uzairkhan@ciit.net.pk

Ihsan Ullah
Ihsan@cuiatd.edu.pk

[1] Department of Electrical Engineering, Mirpur University of Science and Technology (MUST), Mirpur, AJK 10250, Pakistan

[2] Department of Electrical Engineering, CUI Islamabad, Abbottabad Campus, Abbottabad 22060, Pakistan

The snake robot constructed by Hirose's (1993) was one of the first of many such robots that have been produced till now. So far the snake robots are classified into five classes on the basis of their work and design (Hirose and Yamada 2009; Hopkins et al. 2009).

1. The snake robots developed with combination of links and attached to each other with active revolute joints (Crespi and Ijspeert 2006; Wright et al. 2012). If snake robot has passive wheels and active joints, it also falls in this type.
2. The snake robots having collection of links and joints that revolve as well as change their length (Sugita et al 2008; Manzoor and Choi 2016).
3. The snake robot that have both active wheels and active joints in it (Yamada et al. 2005).
4. The snake robot consisting of active wheels, passive joints (Kimura and Hirose 2002).
5. The snake robot having crawlers and active joints (Borenstein et al. 2007; Arai et al. 2008).

Along with the construction, many control algorithms have also been devised for the motion of the snake-like robots (Javaheri et al. 2018). Central pattern generator (CPG)-based motion control (Ijspeert 2008; Junzhi et al. 2014) is an effective solution for the bio-inspired robots.

CPG can also be used for snake robots, since the motion of real snake is rhythmic in nature (Manzoor et al. 2019). Along with this, the control of the redundant snake-like mechanisms is made easy by using CPG. The CPG-based motion of snake is proposed in Ijspeert et al. (2007). In Wu et al. (2010), the authors have generated crawling motion in snake robot by using a CPG that is adaptive to changes in friction and inclination of surface on which it moves. The adaptation to environment using CPG to generate serpentine and side-winding motion in a robot is given in Tang et al. (2010). In Nor and Ma (2014), the authors have given the smooth transition of robot body from stationary to serpentine motion. The CPG is also used to generate rhythmic motions in snake robots in Wang et al. (2017), Zhenshan et al. (2017). However, in each of these researches only single CPG-based motion of snake-like robot is achieved. In Manzoor and Choi (2016), the authors proposed neural oscillator-based algorithm for other types of snake-like motion. However, only the rhythmic motion generator for different types of motions was given, the results in that paper were only generated by using the simulations. The limit cycle for the stable motion of robot was also not discussed in that paper. Furthermore, the mechanism for transition between different types of motions, which is a contemporary research topic in CPG-based research Yu et al. (2016), was not given in the paper.

In this paper, we have proposed a neural oscillator-based CPG, having smooth limit cycle properties, for serpentine and rectilinear motion generation in a newly constructed snake robot. The mechanism for the smooth transition between these two types of motion and transition from stationary position to any type of motion in the robot is also proposed in the paper. A snake robot having passive joints and active wheels is designed and constructed. The robot can bend on the ground as well as the length of snake can be increased and decreased. A layout of experimental setup along with global positioning system is also given in the paper. In this way proposed CPG can be applied using experiments along with the simulations.

The paper is further organized as follows: The structure, kinematics and the hardware of snake robot is given in Sect. 2. The neural oscillator-based CPG, along with the smooth gait transition mechanism is given in Sect. 3. The discussion on the use of CPG algorithm for different kinds of motions generation in the constructed robot is given in Sect. 4. The experiments and then the results in the form of simulations, carried out on snake robot, obtained by using the proposed algorithm are given in Sect. 5. In Sect. 6, the paper is concluded.

2 Constructed Snake Robot

Construction of a snake robot, with passive joints and active wheels, is given in this section. It was first proposed by the author in Manzoor and Choi (2016) and its mechanism is shown in Fig. 1a. As all the joints are with passive actuators, showing pitch rotation, thus it can only be moved on flat ground or on inclined surface. It mainly consists of a chain of links, each with a pair of active wheels separated by length "L". Between ith and $(i+1)$th wheel-link, there are two passive prismatic joints, connected through passive revolute joint, and defined as P_{i_1} and P_{i_2} as given in Fig. 1a. There are $2(m-1)$ passive prismatic and $m-1$ passive revolute joints in the robot for m number of wheel-links. The revolute joints help in the bending of the snake robot on the ground, for serpentine motion, while the passive prismatic joints help in achieving the property of change in the length of the robot for rectilinear motion.

In Fig. 1a, world coordinate frame is denoted by X–Y and the coordinate frame for snake, in line with the movement of snake, is donated by X_S–Y_S, e.g., robot moves along with X_S-axis. Z-axis about which the passive revolute joints rotate, is same for both the coordinate frames. The rotation of the snake coordinate frame to the world coordinate frame is defined as rotation of angle ϑ_S. Let the position of the center point of ith wheel-link in world frame is denoted by (x_i, y_i) and its orientation by ϑ_i, then the kinematics of the proposed mechanism of Fig. 1 can be written as;

$$x_{i+1} = x_i + \left(\frac{L}{2} + P_{i_1}\right)\cos\vartheta_i + \left(\frac{L}{2} + P_{i_2}\right)\cos\vartheta_{i+1} \quad (1)$$

$$y_{i+1} = y_i + \left(\frac{L}{2} + P_{i_1}\right)\sin\vartheta_i + \left(\frac{L}{2} + P_{i_2}\right)\sin\vartheta_{i+1} \quad (2)$$

Let V_{i_1} be linear and V_{i_2} be the angular velocities of ith link, respectively, then the velocity kinematics of body is given as;

$$\dot{x_i} = V_{i_1}\cos\vartheta_i \quad (3)$$

$$\dot{y_i} = V_{i_1}\sin\vartheta_i \quad (4)$$

$$\dot{\vartheta_i} = V_{i_2}. \quad (5)$$

2.1 Hardware

Figure 1b shows passive joints and active wheel snake robot, constructed on the base of the design given in Fig. 1a. It consists of four links with wheel, i.e., "$m = 4$". The length "L" of each wheel-link is 140 (mm). The passive prismatic joints are formed by connecting linear guides to these links. The maximum lengths of the prismatic joints is $P_{i_1}(\max) = P_{i_2}(\max) = P_0 + 35\,(\text{mm})$ and

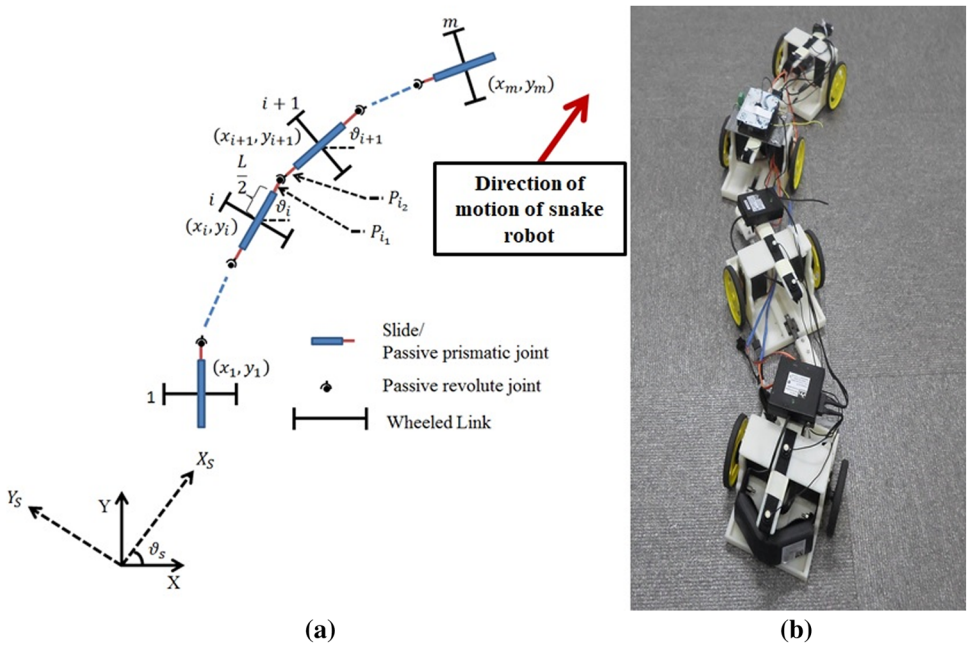

Fig. 1 Snake robot: **a** mechanism, and **b** constructed robot

Table 1 Construction parameters of snake robot with active wheels passive joints

Parameter	Value
Number of wheel-links (m)	4
Number of passive prismatic joints (linear rails)	6
Number of passive revolute joints	3
Length of wheel-link (L)	140 (mm)
Distance between the two wheels	120 (mm)
Maximum length of prismatic joint P_{i_1}(max)	35 (mm)
Side-wise maximum angle ϑ(max)	45°

its minimum length is $P_{i_1}(\min) = P_{(i_2}(\min) = P_0\,(\mathrm{mm})$, where $P_0 = 89\,(\mathrm{mm})$. The active wheels are run by servo motors with continuous rotation. The detailed kinematic parameters for constructed robot are given in Table 1.

Commercially available Arduino Uno is used to control the motors. The robot has no local sensor; thus, a global sensor is required to detect the position (x_i, y_i) of the center of ith link in X–Y plane (world frame) as well as its orientation ϑ_i in the horizontal direction. This is done by using PTI VisualeyezTM Motion-capture System. This system uses the precision wireless markers at the center of each wheel-link. However, it has limited workspace; therefore, it is used for small-scale experiments only. On large scale, GPS can be used in outdoor environment. The screen-short of VzSoft software for the global positioning sensor is shown in Fig. 2. It can be seen that in order to find the orientation ϑ_i of ith link another marker is added on each wheel-link in addition to the one at the center.

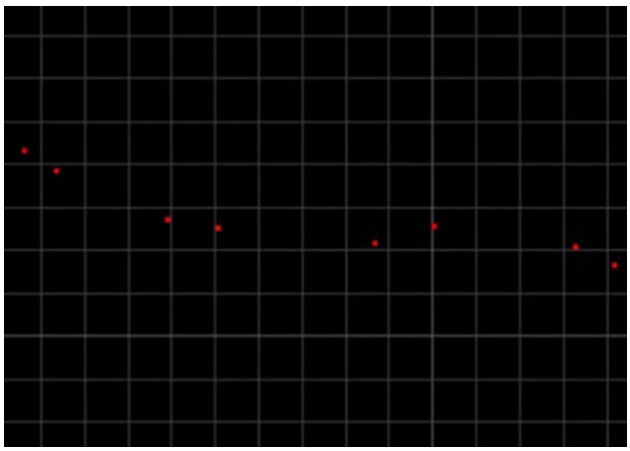

Fig. 2 Snapshot of vzsoft screen-short, showing marker positions in PTI VisualeyezTM Motion-capture System

3 Neural Oscillator-Based Central Pattern Generator

All the modes of locomotions in snake are rhythmic in nature. The serpentine locomotion is a continuous motion during which its move forward making a sinusoidal wave-like motion. On the other hand during the rectilinear motion, the body of the snake moves from tail to the head in such a way that different parts of the snake body move one by one, sequentially. Thus, in this section, we would give a CPG for the rhythmic serpentine and the rectilinear motion for the proposed snake robot.

In order to get the periodic serpentine and rectilinear movement for the constructed snake robot, shown in Fig. 1b, a CPG based on neural oscillator with the

entrainment property can be used. The required neural oscillator should form limit cycles of different shapes for both the serpentine as well as the rectilinear motions. Such a CPG can be formed by Kuramoto oscillator (Sksguchi and Kuramoto 1986) [which was further extended in Ijspeert et al. (2007)]. This oscillator is combined with two first-order radial functions. The proposed central pattern generator consists of coupled neural oscillators with a excitatory neuron and inhibitory neuron for each wheel-link.

Let θ_j and θ_k denote angles of jth and kth neuron, respectively, and $W_{j,k}$ denote the coupling weight between them. Let r_j be the radial amplitude parameter of the desired limit cycle of jth neuron. While "s" be the parameter to control the magnitude and the phase difference $\phi_{j,k}(s)$ between jth and kth neurons, with $s(0)$ being initial value of s and S is desired final value to which "s" changes. Then, for n total number of neurons and $j = 1, 2, \ldots, n$, the model of oscillator for jth neuron is written as

$$\dot{\theta}_j = \omega_j + \sum_{k=1}^{n} W_{j,k} \sin(\theta_k - \theta_j - \phi_{j,k}(s)) \tag{6}$$

$$\dot{r_j} = \alpha_1((s - s(0))f(\theta_j) - r_j) \tag{7}$$

$$\dot{s} = \alpha_2(S - s) \tag{8}$$

where ω_j the natural frequency of jth neuron with condition $\omega_1 = \omega_2 = \cdots = \omega_n = \omega$Izhikevich and Moehlis (2004) for synchronization. Speed of robot can also be changed by varying value of ω_j. Furthermore, the function $f_j(\theta_j)$ in (7) form the shape of the limit cycle. Different functions can be used to generate limit cycle of our choice of shape. For serpentine and the rectilinear types of motions in snake, following sigmoid function $f_j(\theta_j)$ can be used;

$$f(\theta_j) = \frac{1}{1 + e^{b_j - a_j \cos\theta_j}} \tag{9}$$

This function generates continuous sinusoidal firing for serpentine motion and a neuron firing with recovery period for rectilinear motion. a_j and b_j adjusts the recovery period, amplitude and rise time of neuron output. For the serpentine locomotion we take $a_j = 1$ and $b_j = 0$ and for the case of rectilinear motion we take $a_j = b_j + 6$ and $b_j = 40$. In order to get smooth transaction between different types of snake robot motion the phase difference $\phi_{j,k}$ is taken as a function of parameter "s" as follows:

$$\Phi_{j,k} = \Phi_{j,k}(0)(S - s) + \left(\Phi_{j,k} - \Phi_{j,k}(0)\right)(s - s(0)) \tag{10}$$

where $\Phi_{j,k}(0)$ is phase difference at the start of each transition and $\Phi_{j,k}$ is the required phase difference between jth and kth neuron in each of serpentine and rectilinear motion in the neural oscillator. On the other hand, $s(0)$ is the starting value of parameter s. The value of parameter $S = s(0) + 1$ for new mode of motion. For each transition of gait, $s(0)$ is equal to the value of constant parameter S for previous mode of motion.

In Fig. 3, the output of the proposed CPG based on neural oscillator is shown. Only four couple oscillators are used with the neighboring coupled oscillators having final phase difference $\Phi_{j,k} = \pi/2$. In the neural oscillator for sinusoidal output, the coupling excitatory neuron and inhibitory neuron oscillators have final phase difference $\Phi_{j,k} = \pi$, while $\Phi_{j,k} = 0$ for the neural oscillator with the recovery period. The neuron firing in Fig. 3a is continuous and sinusoidal, whereas in Fig. 3b, it is not continuous, i.e., the neuron fire rests and then fires again. The neurons with phase difference $\Phi_{j,k} = 0$ fire simultaneously, while the neuron with phase difference $\Phi_{j,k} \neq 0$, do not fire at the same time.

In Fig. 4, the limit cycles with and without using smoothing Eq. (10) are shown. It is evident from Fig. 4a that the limit cycle converges smoothly from equilibrium and remains nonzero if Eq. (10) is used for the sinusoidal output. While if Eq. (10), is not used the convergence of limit cycle is not smooth, as shown in Fig. 4b. The limit cycle of neurons with recovery period, with consideration and without considering Eq. (10), is shown in Fig. 4c, d and it can be noticed that both almost pass through zero and have similar shapes.

4 Control for Snake-Like Robot Using CPG

In order to control the motion of the snake robot, a network of neural oscillators of Eqs. (6), (7) and (8) is proposed. The output of this network would control the orientation of each wheel-link as well as the length of the prismatic joints. This network as well as the layout and control sequence is shown in Fig. 5. The network has a pair of excitatory and inhibitory neuron for orientation control of each wheel-link while in order to control the length of prismatic joint (joining two consecutive wheel-links) another pair of neural oscillator is added to the network. Thus, for the robot with m wheel-links and $m - 1$ pairs of prismatic joints $2m - 1$ pairs of neurons i.e. $n = 4m - 2$ number of neurons are required. The coupling between jth and kth neuron for orientation control in the network of Fig. 5 can be managed by taking $W_{j,k} = 0$ if no arrow connect them and $W_{j,k} = 1$ if there is an arrow from jth neuron to kth neuron. Same is for the case of $W_{pj,pk}$, the pjth and pkth neurons of the prismatic joints.

It is assumed that the each wheel-link acts as an individual two wheel mobile robot with constraints of rigid body and these links are connected with each other in a

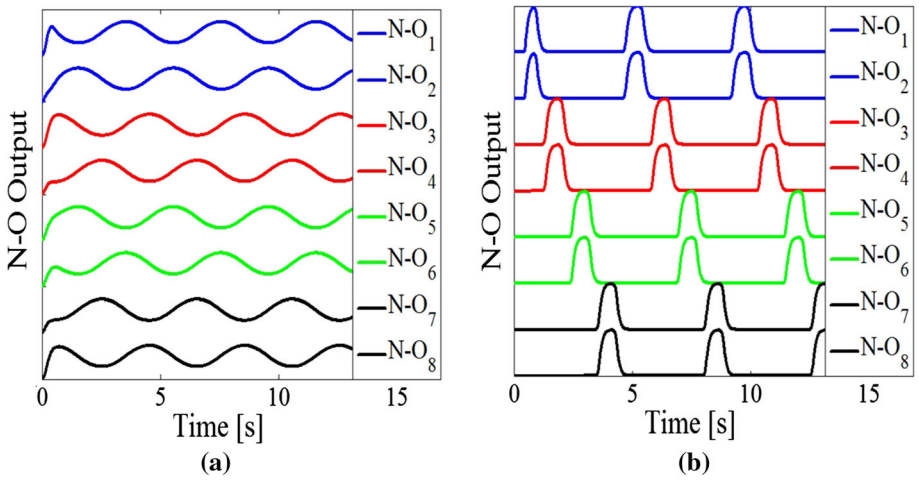

Fig. 3 Neural oscillator (N-O) output **a** neurons with sinusoidal output, **b** neurons with recovery period

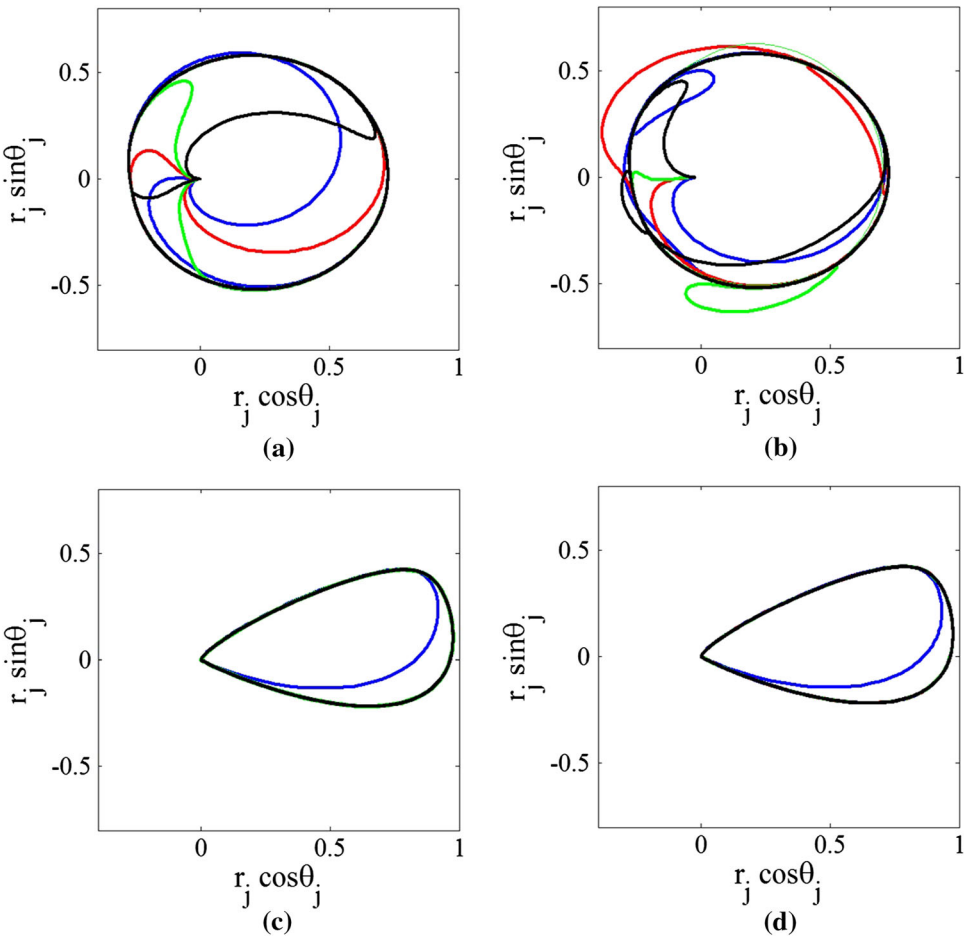

Fig. 4 Neural oscillator-based limit cycle **a** limit cycle with sinusoidal output with Eq. (10), **b** limit cycle with sinusoidal output, without Eq. (10), **c** limit cycle for neurons with recovery period with Eq. (10), and **d** limit cycle for neurons with recovery period without Eq. (10)

chain. The desired orientation and distance between different links can be calculated from the neural activation given in Eq. (9). Let us denote the snake coordinate orientation of ith link of snake robot, by ϑ_{Si}. The sum of the desired length of two prismatic joints between ith and $(i+1)$th link can be written as d_{pi}. Furthermore, the desired angle of ith link denoted by $\vartheta_{S_{di}}$ can be generated from following relations;

$$\vartheta_{S_{di}} = \lambda_i(r_{j+1} - r_j) \tag{11}$$

where $j = 2i - 1$ for $i = 1, 2, \ldots, m$ with $m = 4$, and λ_i is to adjust the amplitude value of desired ϑ_{di} as multiples of .4575 (rad).

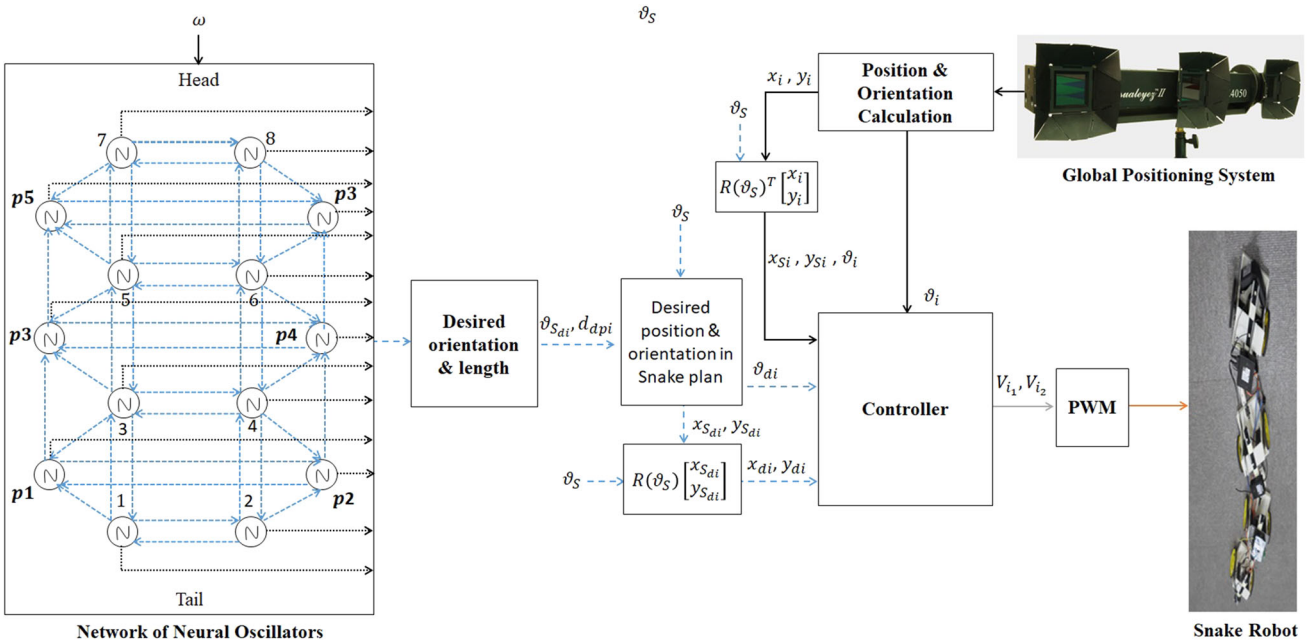

Fig. 5 Control strategy for proposed algorithm

Furthermore, the desired length of prismatic joints located between the ith and $(i+1)$th wheel-link is generated as follows:

$$d_{\text{dpi}} = \beta\left(1 - \frac{r_{dj+1} + r_{dj}}{2}\right) \tag{12}$$

where parameter β is selected according to the sum of maximum distance that the two prismatic joints can traveled, i.e., $P_{i_1}(\max) + P_{i_2}(\max)$. The value of d_{dpi} obtained from Eq. (9) is so adjusted that the desired length of prismatic joints does not exceed its maximum value, i.e.,

$$P_{i_1}(\min) + P_{i_2}(\min) \le 2P_0 + d_{\text{dpi}} \le P_{i_1}(\max) + P_{i_2}(\max)$$

Let we denote the measured horizontal position, of first wheel-link, in snake plane as x_{S1} and initial value of vertical snake plane in required direction as $y_S(0)$, then the desired center position of the tail wheel-link (i.e. $i = 1$) is given in snake plane as;

$$x_{Sd1} = x_{S1} + \zeta_{xs}\cos\vartheta_{Sd1} \tag{13}$$

$$y_{Sd1} = y_S(0) + \zeta_{ys}\sin\vartheta_{Sd1} \tag{14}$$

where the constants ζ_{xs} and ζ_{ys}, respectively, controls the motion of first wheel-link in horizontal and vertical snake plane. On the other hand, the desired position (x_{Sdi}, y_{Sdi}) of ith wheel-link with $i > 1$ can be given as

$$x_{Sdi} = x_{S(i-1)} + \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\cos\vartheta_{Sd(i-1)} + \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\cos\vartheta_{Sdi} \tag{15}$$

$$y_{Sdi} = y_{S(i-1)} + \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\sin\vartheta_{Sd(i-1)} + \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\sin\vartheta_{Sdi} \tag{16}$$

The robot would move forward using the relations given above. However, Eqs. 15 and 16 will become

$$x_{Sd(i-1)} = x_{Sdi} - \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\cos\vartheta_{Sd(i-1)} - \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\cos\vartheta_{Sdi} \tag{17}$$

$$y_{Sd(i-1)} = y_{Sdi} - \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\sin\vartheta_{Sd(i-1)} - \left(\frac{L}{2} + P_0 + \frac{d_{dp(i-1)}}{2}\right)\sin\vartheta_{Sdi} \tag{18}$$

if condition $|atan2(y_{Sdi} - y_{Si}, x_{Sdi} - x_{Si})| > \pi/2$ is met.

The desired position (x_{di}, y_{di}) in the world coordinate can be obtained from the desired position (x_{Sdi}, y_{Sdi}) in the snake frame, which is at a rotation angle ϑ_S w.r.t. world coordinate, by using following relation

$$\begin{bmatrix} x_{di} \\ y_{di} \end{bmatrix} = \begin{bmatrix} \cos(\vartheta_S) & -\sin(\vartheta_S) \\ \sin(\vartheta_S) & \cos(\vartheta_S) \end{bmatrix} \begin{bmatrix} x_{S_{di}} \\ y_{S_{di}} \end{bmatrix} \quad (19)$$

The desired angle of the ith wheel-link in the world coordinate is given as:

$$\vartheta_{di} = \vartheta_{S_{di}} + \vartheta_S \quad (20)$$

4.1 Transition Between the Different Modes of Motion

For smooth transition of gait from stationary to serpentine, stationary to rectilinear, serpentine to rectilinear and vice versa, following steps are taken;

1. Firstly the parameter S is changed in Eq. (8) such that $S = s(0) + 1$. With $s(0)$ being the value of parameter "S" in previous mode of motion.
2. The value of phase difference $\Phi j, k$ at the start of transaction is taken as $\Phi j, k(0)$ and new value of $\Phi j, k$ is chosen according to the another type of gait as suggested in "Appendix".
3. In case of transaction from stationary position to serpentine, or rectilinear motion to serpentine motion, the values of parameters a and b are changed when parameters $\Phi j, k$ and S are changed.
4. In case of transaction from stationary position to rectilinear, or serpentine to rectilinear motion, the values of parameters a and b are changed when $s \approx S$.

5 Results

5.1 Simulation Results

The proposed algorithm is first applied in simulations by creating snake robot in MATLAB environment. The graphical functions of MATLAB are used to simulate the snake robot. The constraints of revolute and prismatic joints, as well as non-holonomic constraints of wheel-links are also added in the MATLAB program. The algorithm given in Sect. 4 is used to calculate desired wheel-link angles and distance between two consecutive links. In order to get better understanding of the proposed algorithm, we have chosen $m = 8$ wheel-links, which is unlike the real robot is given in Sect. 2, with $m = 4$. The Length "L" of every wheel-link and distance between the two wheels in a links are all set as 1 (unit). On the other hand, we take $P_{i_1}(\max) = P_{i_2}(\max) = 0.5$ (unit). For simplicity we take $\vartheta_S = 0$ and $\omega = \frac{\pi}{2}$.

Firstly, the robot, in the simulation, is moved from stationary position to serpentine type of locomotion. Then, the mode of motion is changed, form serpentine to rectilinear type of locomotion. Finally, the motion is again transformed to the serpentine type of motion. In order to get serpentine motion, we take $a = 0$ and $b = 1$, and for rectilinear motion $a = 40$ and $b = 43$. The values of parameter $\Phi j, k$ for serpentine motion for one full sine wave and the rectilinear motion are taken according to the "Appendix".

The snake robot motions are generated by using neuron firing obtained from CPG proposed in Sect. 3. The desired orientation of each wheel-link and the length of the prismatic joint are obtained by algorithm given in Sect. 4. The changes in value of parameter "s", to obtain different motions, are shown in Fig. 6a. Initially, at stationary position, we have taken $s = 0$, then its value is changed to 1 for serpentine motion. The value of parameter "s" is taken to be 2 for transition from serpentine to rectilinear motion. Then, it is changed to 3 for transition from rectilinear to serpentine motion. In Fig. 6b, the neuron firings for the orientation control of the wheel-link are shown. It can be observed that, in the serpentine locomotion, after some transient region, the neurons arrange themselves at a phase difference $\frac{2\pi}{m}$, although they started with the same initial values. They continuously rise and fall in a clamped sine wave shape. On the other hand, in rectilinear motion, these neurons remain at "0" phase differences, and each of them sequentially rises to maximum value then reduced to approximately zero value. In Fig. 6c, the neuron firings that generate the desired length of prismatic joints are shown. In the period of serpentine locomotion, all neuron outputs start from same value, at the end the phase difference remain "0" and amplitude of each output remains same. In Fig. 7a, the respective orientation of each wheel-link for snake motion is given. During serpentine motion, orientation is oscillatory while it remains at a constant value of "0" (rad), in the rectilinear motion. In Fig. 7b, the respective sum of length of prismatic joints located between each wheel-link is given. It can be observed that the desired length of prismatic joint remains constant during the serpentine locomotion, while it sequentially changes, one by one from head to tail, during rectilinear motion.

Several snapshots of the MATLAB simulation of snake robot locomotion are shown in Fig. 8. During the serpentine motion [Fig. 8 (3–5 and 19, 20)], it can be seen that robot shows a continuous motion, forming sine wave-like shape. While during the rectilinear type of locomotion [Fig. 8 (7–15)], each wheel-link of the robot moves one by one from tail to head. It can be noticed that the transaction of motion from stationary to serpentine [Fig. 8 (1, 2)], serpentine to rectilinear [Fig. 8 (6, 7)] and then again from rectilinear to the serpentine [Fig. 8 (16–18)] motion is smooth.

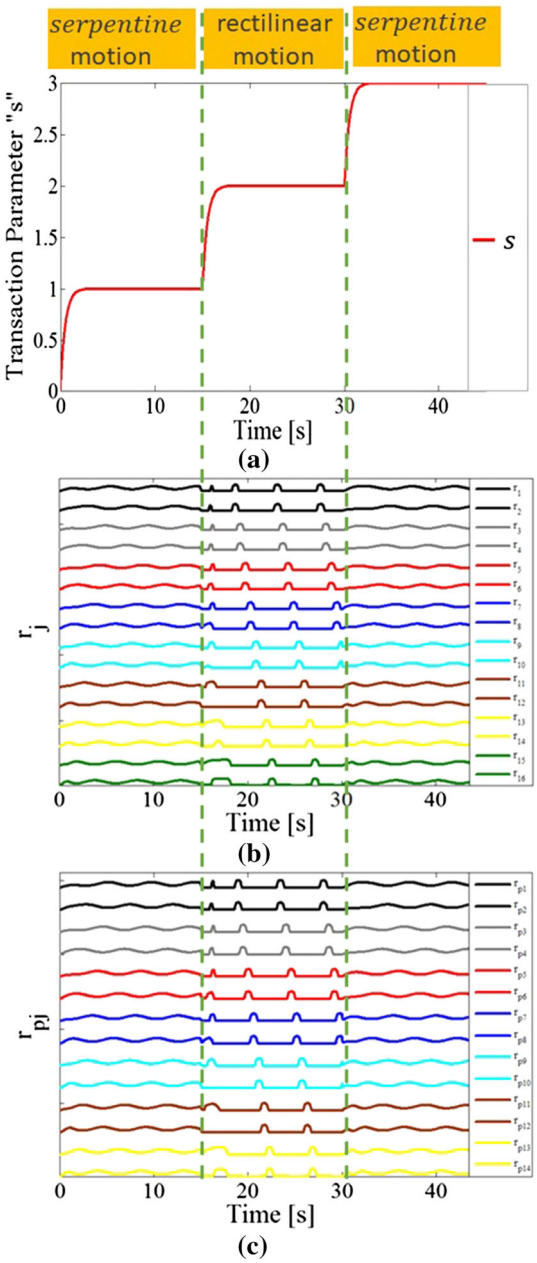

Fig. 6 **a** Parameter "*s*" during the different types of snake motions and transitions between them, **b** neuron firing for orientation of revolute joints, and **c** neuron firing for length of prismatic joints

5.2 Experimental Results

The experiments on CPG-based control are conducted using the snake robot given in Fig. 1b. Global positioning using VZSoft software is used to estimate the position of each wheel-link. Two markers are used for each wheel-link. The marker at the center position of each wheel-link is used to get its position in the global plane, while the second marker is used to estimate the orientation of link in the world frame. The original length of each prismatic joint is

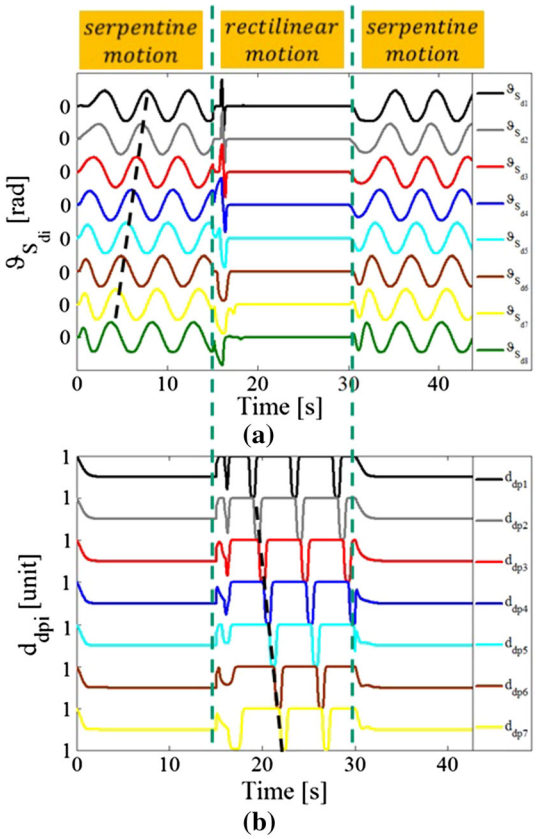

Fig. 7 Output of CPG; **a** orientation of revolute joints, and **b** length of prismatic joints

calculated from actual position and orientation of consecutive wheel-links. The estimated position and orientation of each wheel-link is transferred to commercially available Arduino controller using serial communication. Here, the desired orientation and displacement is obtained using proposed CPG-based algorithm.

In the first experiment, the robot is moved from stationary position to the serpentine type of locomotion. The rough surface is used in order to avoid slip in the robot. The sequential camera snapshots of robot and respective screenshots of vzsoft, showing marker positions in PTI VisualeyezTM Motion-capture System, for serpentine motion, are given in Fig. 9a. The robot body forms a continues sine wave shape during motion. In another experiment, the robot is moved from stationary position to the rectilinear type of locomotion using the proposed CPG algorithm. The sequential camera snapshots for this types of motion, along with respective screenshots of vzsoft are given in Fig. 9b. Each wheel-link of the robot moved from head to tail, slowly and sequentially. It can be noticed that the snake moves in straight line, while each wheel-link, moves, one by one, from tail to head. In this way, the robot moves forward.

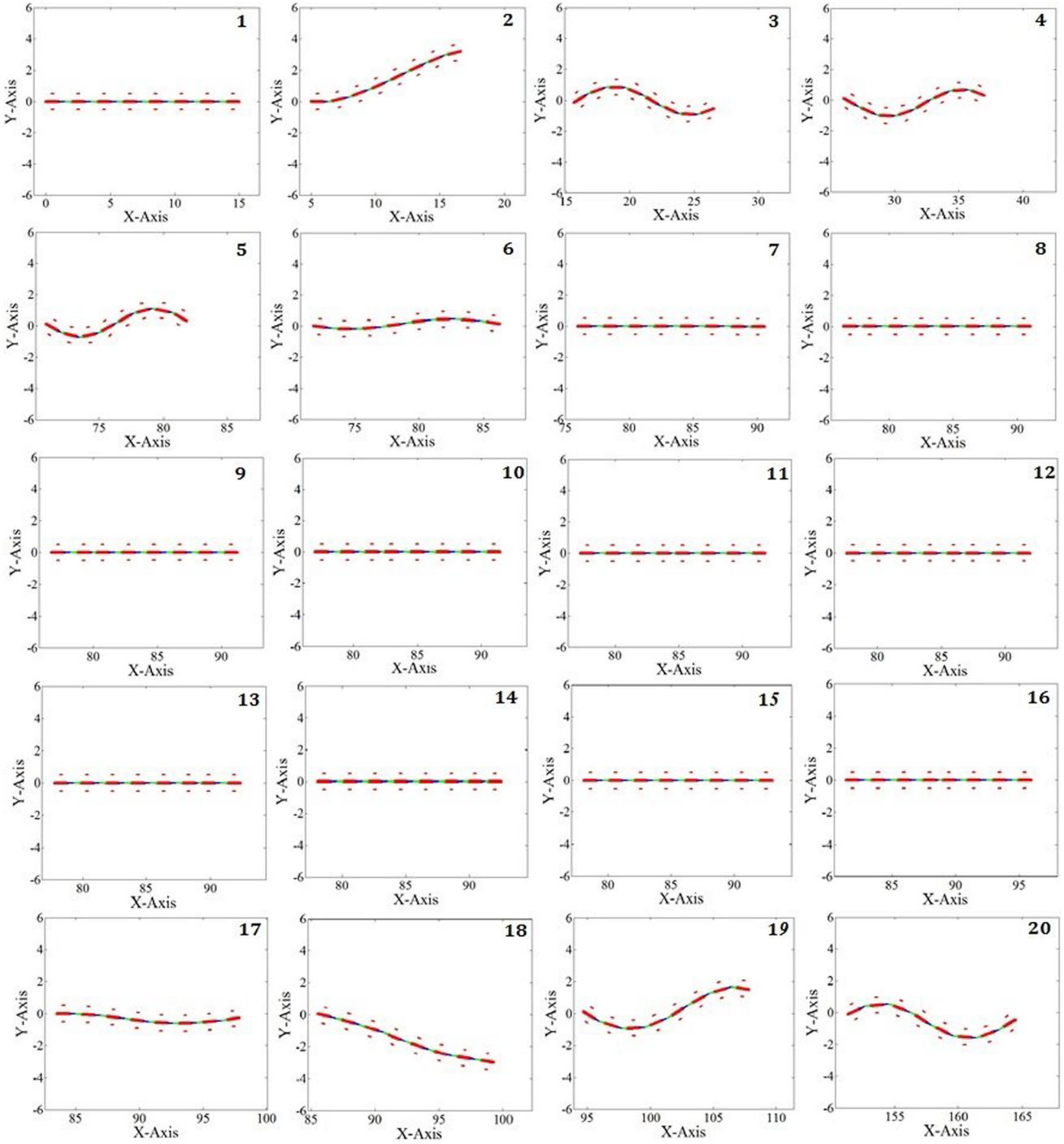

Fig. 8 Snapshots of simulations for stationary-serpentine-rectilinear-serpentine transition motion in snake

6 Concluding Remarks

The construction of a snake robot with active wheels and passive joints, along with formulation of CPG-based algorithm to generate rhythmic motions in it, is proposed in this paper. The robot consists of a four connected links each having two wheels a prismatic and a revolute joint. The passive prismatic joints help in changing the size of the snake robot, while the revolute joints help in the bending of its body. The snake robot can only move on ground. The proposed CPG could generate both the serpentine as well as rectilinear motions along with smooth transition between these two motions as well as motion from stationary position can also be achieved. Through simulation

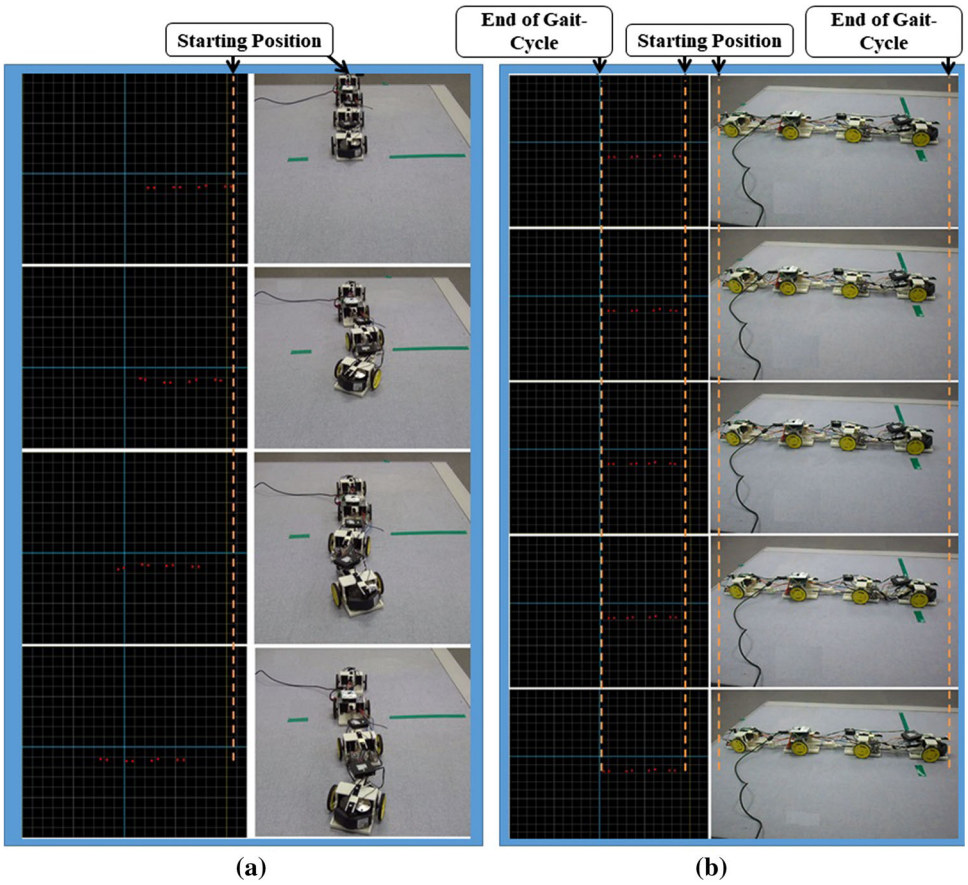

Fig. 9 Snapshots of experiments on snake-like robot; **a** serpentine motion, **b** rectilinear motion

studies and experiments, the effectiveness of proposed algorithm is verified. During the serpentine motion, a continuous motion of snake robot is achieved where its body forms the shape of sine wave while during the rectilinear motion the snake robot body gradually moves from tail to head.

Acknowledgements This work was supported in part by the Higher Education Commission, Pakistan under "Startup Research Grant Program" No: 21-2064/SRGP/R&D/HEC/2018. The authors are also thankful to Prof. Dr. Youngjin Choi, Biorobotics Lab, Hanyang University South Korea, for his support and provision of equipments for the experiments.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix: Phase Difference

1. The phase difference $\Phi_{j,k}$ in the serpentine motion is selected so that the desired orientation of the wheel-links should be sinusoidal and the length of each prismatic joint remain constant. This would generate continuous, nonnegative, sinusoidal neuron firing. To achieve these requirements, the coupled neurons of the same wheel-link and prismatic joint are taken out of phase (e.g., $\Phi_{j,k} = \pm\pi$);

$$\Phi_{j,k} = \pm\pi \quad \forall \quad j = k \tag{21}$$

While the phase difference between the neurons of adjacent wheel-links are set such that

$$\Phi_{j,k} = \pm\frac{2n\pi}{m} \quad \forall \quad j \neq k \tag{22}$$

where n is the total number of required sine waves for robot. Thus, the phase difference between the neurons of adjacent wheel-links should be equal to $\Phi_{j,k} = \pm\frac{2\pi}{m}$ for one full sine wave.

2. In the rectilinear locomotion, the orientation of all the wheels should be same, along the horizontal direction of snake coordinate. The length of the prismatic joints should alternatively change from tail to head (wheel-links should move alternatively). This can be achieved such that the neuron of same coupled oscillators should fire simultaneously, i.e., $\Phi_{j,k} = 0$.

$$\Phi_{j,k} = \pm 0 \quad \forall \quad j = k \tag{23}$$

Along with phase differences between neurons of the adjacent wheel-links are taken such that

$$\Phi_{j,k} = \pm\frac{2\pi}{m} \quad \forall \quad j \neq k \tag{24}$$

in this way total sum of all phase difference becomes 2π.

References

Arai M et al (2008) Development of "Souryu-IV" and "Souryu-V": serially connected crawler vehicles for in-rubble searching operations. J Field Robot 25(1–2):315–322

Borenstein J et al (2007) The omnitread OT-4 serpentine robot-design and performance. J Field Robot 24(7):601–621

Crespi A, Ijspeert AJ (2006) AmphiBot II: an amphibious snake robot that crawls and swims using a central pattern generator. In: International conference on climbing and walking robots, Brussels, pp 19–27

Hirose S (1993) Biologically inspired robots: snake-like locomotors and manipulators, vol 1093. Oxford University Press, Oxford

Hirose S, Yamada H (2009) Snake-like robots machine design of biologically inspired robots. IEEE Robot Autom Mag 16(1):88–98

Hopkins JK et al (2009) A survey of snake inspired robot designs. Bioinspir Biomim. https://doi.org/10.1088/1748-3182/4/2/021001

Ijspeert AJ et al (2007) Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In: IEEE international conference on robotics and automation, pp 262–268

Ijspeert AJ (2008) Central pattern generators for locomotion control in animals and robots: a review. Neural Netw 21(4):642–653

Izhikevich EM, Moehlis J (2004) Dynamical systems in neuroscience: the geometry of excitability and bursting. Springer, New York

Javaheri KM et al (2018) Bio-inspired snake robots: design, modeling, and control. In: Habib M (ed) Handbook of research on biomimetics and biomedical robotics, IGI global, pp 246–275

Junzhi Y et al (2014) A survey on CPG-inspired control models and system implementation. IEEE Trans Neural Netw Learn Syst 25(3):441–456

Kimura H, Hirose S (2002) Development of Genbu: active wheel passive joint articulated mobile robot. In: EEE/RSJ international conference on intelligent robots and system, pp 823–828

Manzoor S, Choi Y (2016) A unified neural oscillator model for various rhythmic locomotions of snake-like robot. Neurocomputing 173(Part 3):1112–1123

Manzoor S, Cho YG, Choi Y (2019) Neural oscillator based CPG for various rhythmic motions of modular snake robot with active joints. J Intell Robot Syst 94(Part 3–4):641–654

Manzoor S, Choi Y (2016) Modular design of snake robot for various motion implementation. In: 13th international conference on ubiquitous robots and ambient intelligence (URAI), pp 211–213

Nor NM, Ma S (2014) Smooth transition for CPG-based body shape control of a snake-like robot. Bioinspir Biomim 9:016003

Sksguchi H, Kuramoto Y (1986) A soluble active rotator model showing phase transitions via mutual entrainment. Int Symp Math Probl Theor Phys Lect Notes Phys 76(3):576–581

Sugita S et al (2008) A study on the mechanism and locomotion strategy for new snake-like robot active cord mechanism Slime model 1 ACM-S1. J Robot Mechatron 20:302–310

Tang C et al (2010) A cubic CPG model for snake-like robot to adapt to environment. In: IEEE international conference on information and automation (ICIA), pp 24–29

Wang Z et al (2017) CPG-inspired locomotion control for a snake robot basing on nonlinear oscillator. J Intell Robot Syst 85(2):209–227

Wright C et al (2012) Design and architecture of the unified modular snake robot. In: IEEE international conference on robotics and automation, vol 36, no 4, pp 425–443

Wu X, Ma S et al (2010) Adaptive creeping locomotion of a CPG-controlled snake-like robot to environment change. Auton Robots 28(3):283–294

Yamada H et al (2005) Development of amphibious snake-like robot ACM-R5. In: International symposium on robotics, pp 617–624

Yu H et al (2016) Gait generation with smooth transition using CPG-based locomotion control for hexapod walking robot. IEEE Transit Ind Electron 63(9):5488–5500

Zhenshan B et al (2017) Towards autonomous locomotion: CPG-based control of smooth 3D slithering gait transition of a snake-like robot. Bioinspir Biomim 12(3):035001