Intel Serv Robotics (2013) 6:109–120
DOI 10.1007/s11370-013-0129-3

ORIGINAL RESEARCH PAPER

Development of a new spinning gait for a planar snake robot using central pattern generators

Shahir Hasanzadeh · Alireza Akbarzadeh

Received: 1 June 2012 / Accepted: 22 January 2013 / Published online: 7 March 2013

Abstract In this paper, we first present dynamic equation of n-link snake robot using Lagrange's method in a simplified matrix form and verify them experimentally. Next, we introduce a new locomotion mode called spinning gait. Central pattern generators (CPGs) are used for online gait generation. To realize spinning gait, genetic algorithm is used to find optimal CPG network parameters. We illustrate both theoretically, using derived robot dynamics and experimentally that the CPG-based online gait generation method allows continuous and rather smooth transitions between gaits. Lastly, we present an application where the snake robot is guided from an initial to final position while avoiding obstacles by changing CPG parameters.

Keywords Snake robot · Central pattern generator · Serpentine gait · Dynamic · Genetic algorithm

1 Introduction

Snake robots are serially connected, multilink-articulated mechanisms, which propel themselves by body shape undulations. One of the first known biologically inspired snake robots was built by Hirose and co-workers at the end of 1972 [1]. His robot used wheels to create a no side-slip condition for each of the links. A survey of snake robot designs was presented by Hophkins [2].

Snake robots can move with different modes of locomotion (gaits). Serpentine, concertina and side-winding are the three common snake-like gaits in snake robots [3]. There are also non- snake-like gaits which do not exist in nature but are useful in snake robot motion. Flapping gait [4] is an example of non-snake-like gaits. More recently, a novel gait, called FHS, that keeps the head link of the robot towards its target, was introduced by the authors [5].

Modeling and control of snake robot have been addressed by many researchers. A comprehensive survey on snake robot modeling and locomotion has been performed by [6]. In general, control methods may be classified into two broad classes. Sine wave trajectory tracking [7] and online trajectory generation [8]. Recent studies use central pattern generators, CPGs, to generate desired trajectories which can be used by an online controller. CPGs are neural circuits found in both invertebrate and vertebrate animals that can produce rhythmic patterns of neural activity without receiving rhythmic inputs. CPGs present several interesting properties including distributed control, the ability to deal with redundancies, fast control loops, and allowing modulation of locomotion by simple control signals. These properties, when transferred to mathematical models, make CPGs interesting building blocks for locomotion controllers in robots. There are many researches who have realized control of animal-like robots based on CPG model. A good review of application of CPGs in locomotion control of robots can be found in [9].

In the area of snake robots, Crespi et al. [10] proposed CPG-based controller for amphibious snake-like robot and constructed an experimental model. Ma et al. [11,12] proposed different control architectures for serpentine locomotion based on CPG network and simulated them considering mechanical dynamics of a snake robot. More recently, Ryu et al. [13] proposed a CPG-based control architecture which was able to adapt the motion of the robot to varying coefficients of body-ground friction. In all of these works, CPGs were used to change the parameter of a specific gait. In the present paper, we aim to study the capability of CPGs in

S. Hasanzadeh · A. Akbarzadeh (✉)
Center of Excellence on Soft Computing and Intelligent Information Processing, SCIIP,
Mechanical Engineering Department,
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: ali_akbarzadeh_t@yahoo.com

producing smooth transitions between different gaits and apply it to multimodal locomotion of snake robot. In particular, we try to contribute to answer the following questions: (1) how different gaits can be generated for different purposes, and (2) how a smooth transition between different gaits can be achieved.

The goal of this research is to contribute to generate locomotion that more closely imitates real snakes in nature by allowing the robot the choice of a new gait as well as using CPGs as its motion generator which also used by real snakes. To do this, in Sect. 2, we drive the dynamic equation for an n-link snake robot using coulomb friction model. This model will enable us to run our simulation and perform gait parameter optimization. To improve locomotion, we introduce a new gait called spinning gait in Sect. 3. This gait allows the robot to turn with a minimum radius. This feature enables snake robot to improve its maneuverability in environment full of obstacles. A quick and smooth transition between gaits is a difficult problem that is merely addressed in literature. In Sect. 4 and 6, we demonstrate both theoretically and experimentally that CPGs may be used to overcome this problem by allowing a continuous and a smooth transition between serpentine and spinning gaits.

2 Snake robot model

A planar snake robot consisting of n links connected through $n-1$ joints is depicted in Fig. 1a. Each link is rigid with uniformly distributed mass and is equipped with a torque actuator (motor). Moment of inertia of the motors is neglected. Each link is of mass m_i, length l_i and moment of inertia J_i. Let (x_{ci}, y_{ci}) and θ_i define the center of gravity and the angle between the link and the x axis, respectively. Values of d_i represent location of mass center of ith link. (x_b, y_b) is coordinate of the end of tail link.

Free-body diagram of the robot is depicted in Fig. 1b, where T_i are the joint torques from the actuators, and f_{ni} and f_{ti} are the force due to the friction between the links and the horizontal surface. As illustrated in Fig. 1b, φ_i, $(i = 1, \ldots, n-1)$ are relative angles of two adjacent links.

To generate significantly different friction coefficients in normal and tangential directions, a wheel or a blade may be attached to each link. We consider a simple coulomb friction model to simulate friction between each link and ground:

$$f_{ei} = -m_i g \mu_e \text{sign}(v_i^e), \tag{1}$$

where $e = t, n$ (t and n represents tangential and normal directions). g is the gravity constant. μ_t and μ_n are normal and tangential coulomb friction coefficients. Subscript i corresponds to the ith link, f_{ti} and f_{ni} are friction forces in tangential and normal directions, respectively. v_{it} and v_{in} are velocities of the center of mass of ith link. The signum function is denoted by sign(x). The friction forces along the x and y axis can easily be obtained by coordinate transformation from t-n to x-y. According to Fig. 1a, the coordinates of the mass center of ith link are

$$y_{ci} = y_b + \sum_{j=1}^{i-1} l_j \sin\theta_j + d_i \sin\theta_i \tag{2}$$

$$x_{ci} = x_b + \sum_{j=1}^{i-1} l_j \cos\theta_j + d_i \cos\theta_i \tag{3}$$

Velocities of mass center of each link are obtained by taking derivation from Eqs. 2 and 3. Thus, the kinetic energy of the n-link snake robot can be defined as

$$K = \sum_{i=1}^{n} \left[\frac{1}{2} I_i \dot{\theta}_i^2 + \frac{1}{2} m_i (\dot{x}_c^2 + \dot{y}_c^2) \right] \tag{4}$$

Substituting derivatives of Eqs. 2 and 3 into Eq. 4, and summarizing will result in

$$\begin{aligned} K = & \sum_{i=1}^{n} \left[\frac{1}{2} (I_i + m_i d_i^2) \dot{\theta}_i^2 + \frac{1}{2} m_i (\dot{x}_b^2 + \dot{y}_b^2) \right] \\ & + \sum_{i=1}^{n} \left\{ m_i d_i \dot{\theta}_i \sum_{j=1}^{i-1} \left[l_j \dot{\theta}_j \cos(\theta_i - \theta_j) \right] \right\} \\ & + \sum_{i=1}^{n} \left[m_i d_i \dot{\theta}_i (\dot{y}_b \cos\theta_i - \dot{x}_b \sin\theta_i) \right] \\ & + \sum_{i=1}^{n} \left\{ m_i \sum_{j=1}^{i-1} \left[l_j \dot{\theta}_j (\dot{y}_b \cos\theta_j - \dot{x}_b \sin\theta_j) \right] \right\} \\ & + \sum_{i=1}^{n} \left\{ \frac{1}{2} m_i \left[\sum_{j=1}^{i-1} (l_j \sin\theta_j \dot{\theta}_j) \right]^2 \right. \\ & \left. + \frac{1}{2} m_i \left[\sum_{j=1}^{i-1} (l_j \cos\theta_j \dot{\theta}_j) \right]^2 \right\} \end{aligned} \tag{5}$$

The instantaneous system configuration will be known upon having (x_b, y_b) and θ_i $(1 \leq \text{i} \leq \text{n})$. Therefore, the generalized coordinates are selected as follows

$$q_j = [\theta_1, \theta_2, \ldots, \theta_n, x_b, y_b] \tag{6}$$

As there is no variation in potential energy, the equations of motion can be written as

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) + \frac{\partial K}{\partial q_i} - Q_{q_i} = 0 \, (i = 1, 2, \ldots n+2) \tag{7}$$

Non-conservative forces that do work when generalized coordinates are given virtual displacements are actuators

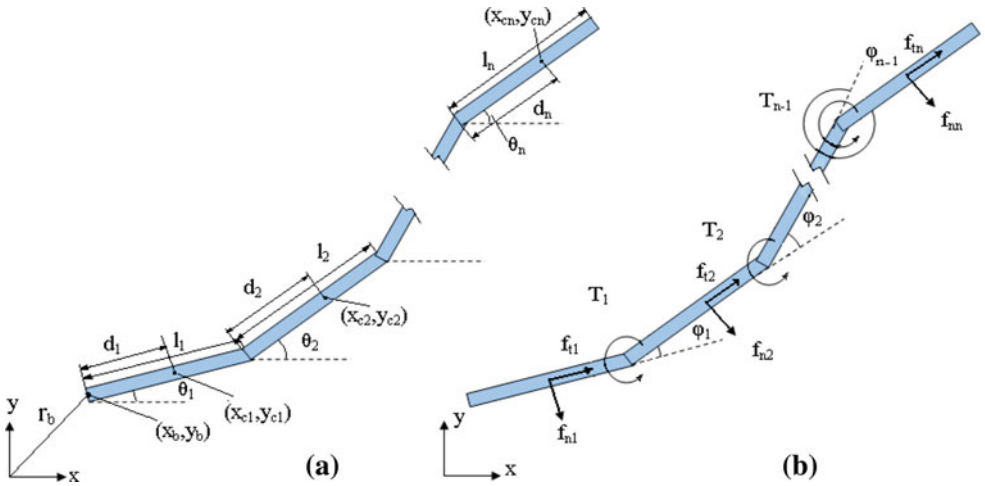

Fig. 1 n-link snake robot: geometrical parameters (**a**), free body diagram (**b**)

torques ($T_i\ i = 1, 2, \ldots, n-1$) and friction forces ($f_{xi}$ and $f_{yi}\ i = 1, 2, \ldots, n$). Taking derivative from Eqs. 2 and 3, generalized forces, Q_{q_i}, can be obtained as

$$\begin{aligned} Q_{\theta j} &= d_j(f_{yi}\cos\theta_j - f_{xi}\sin\theta_j) \\ &\quad + l_j\left[\cos\theta_j \sum_{i=j+1}^{n}(f_{yi}) - \sin\theta_j \sum_{i=j+1}^{n}(f_{xi})\right] \\ &\quad + T_{j-1} - T_j \end{aligned} \tag{8}$$

$$Q_{xb} = \sum_{i=1}^{n}(f_{xi}) \tag{9}$$

$$Q_{yb} = \sum_{i=1}^{n}(f_{yi}), \tag{10}$$

where $Q_{\theta j}$ are generalized forces related to generalized coordinate θ_j.Q_{xb} and Q_{yb} are generalized forces related to x_b and y_b, respectively. By substituting Eqs. 5, 8, 9 and 10 into Lagrangian formulation, Eq. 7, the dynamic model for the n-link snake robot can be derived as

$$BT = M(\theta)\ddot{q} + H(\theta, \dot{\theta}) + F(\theta) \tag{11}$$

where $M(\theta)$ is the $(n+2)\times(n+2)$ positive definite and symmetric inertia matrix, $H(\theta, \dot{\theta})$ is the $(n+2)\times 1$ matrix related to centrifugal and Coriolis terms with H_i as its ith member, $F(\theta)$ is an $(n+2)\times 1$ matrix related to friction forces, B is an $(n+2)\times(n-1)$ constant matrix.T is $(n-1)\times 1$ matrix of input torques and $q, \dot{q}, \ddot{q}$ are $(n+2)\times 1$ matrix of generalized coordinates and their derivatives. θ, $\dot{\theta}$, $\ddot{\theta}$ are $n\times 1$ matrix of links absolute angles and their derivatives. The details of the terms M, H, B and F used in Eq. 11 are presented in Appendix. The final dynamic equation, Eq. 11, has a simplified matrix format and can easily be expanded for any number of links.

2.1 Torque to motion

The forward dynamic problem deals with finding motion of the snake robot, while input joint torques are given. To solve the forward dynamic problem, we simply solve Eq. 11 using Euler method. Equation 11 is an $(n+2)$-dimensional linear equation of $(n+2)$ unknown variables ($\ddot{q} \in R^{n+2}$). By solving this equation, we can obtain angular acceleration for all links ($\ddot{\theta} \in \mathrm{R}^n$) as well as the acceleration of a point at the end of tail link ($\ddot{x}_b$, $\ddot{y}_b$). By integration, links angular velocities ($\dot{\theta}$), joint angles (θ), position (x_b, y_b) and velocity ($\dot{x}_b$, $\dot{y}_b$) of a point at the end of tail link can all be obtained. Therefore, snake robot motion is derived for when input torques for all joints are supplied.

2.2 Shape to motion

In this section, we attempt to drive at the motion of the snake robot given the relative angles of the adjacent links. Another words, given instantaneous relative angles and their derivatives ($\varphi, \dot{\varphi}, \ddot{\varphi}$), Eq. 11 can be solved to find applied torques and coordinates of the tail of the robot which will determine the motion of the robot. Relation between absolute value and relative value of joint angles is:

$$\varphi_i = \theta_{i+1} - \theta_i, \tag{12}$$

where $i = 1, 2, \ldots, n-1$. We can rewrite Eq. 12 in matrix form as,

$$\theta = E\varphi + e\theta_1, \tag{13}$$

where φ is an (n-1)-dimensional vector of $[\varphi_1, \varphi_2, \ldots, \varphi_{n-1}]$, θ_1 is the absolute angle of the tail link, E_{ij} and e are defined as,

$$E_{ij} = \begin{cases} 1 & i > j \\ 0 & \text{others} \end{cases} \quad e = [1, 1, \ldots 1]^T \tag{14}$$

To obtain motion based on given shape, we first decouple the dynamic Eq. 11 into two parts

$$ {}^{p}M(\theta)\ddot{\theta} + {}^{p}N(\theta)\ddot{r}_b + {}^{p}H(\theta,\dot{\theta}) + {}^{p}f(\theta) = DT \quad (15) $$

$$ {}^{q}M + (\theta)\ddot{\theta} + {}^{q}N\ddot{r}_b + {}^{q}H(\theta,\dot{\theta}) + {}^{q}f(\theta) = 0, \quad (16) $$

where

$$ M = \begin{bmatrix} {}^{p}M_{n\times n} & {}^{p}N_{n\times 2} \\ {}^{q}M_{2\times n} & {}^{q}N_{2\times 2} \end{bmatrix}, \quad H = \begin{bmatrix} {}^{p}H_{n\times 1} \\ {}^{q}H_{2\times 1} \end{bmatrix}, \quad r_b = \begin{bmatrix} x_b \\ y_b \end{bmatrix} \quad (17) $$

Matrix ${}^{p}f$, ${}^{q}f$ and D are defined in Appendix. By substituting second derivative of Eq. 13 into Eq. 16, we arrive at,

$$ \begin{aligned} \ddot{r}_b &= -{}^{q}N^{-1}({}^{q}M\ddot{\theta} + {}^{q}H\dot{\theta} + {}^{q}f) \\ &= -{}^{q}N^{-1}{}^{q}M(E\ddot{\phi} + e\ddot{\theta}_1) - {}^{q}N^{-1}({}^{q}H\dot{\theta} + {}^{q}f) \end{aligned} \quad (18) $$

Substituting Eq. 18 into Eq. 15, we obtain

$$ \begin{aligned} DT + ({}^{p}N{}^{q}N^{-1}{}^{q}M - {}^{p}M)e\ddot{\theta}_1 = ({}^{p}M - {}^{p}N{}^{q}N^{-1}{}^{q}M)E\ddot{\phi} \\ - {}^{p}N{}^{q}N^{-1}({}^{q}H\dot{\theta} + {}^{q}f) \\ + {}^{p}H\dot{\theta} + {}^{p}f \end{aligned} \quad (19) $$

Equation 19 is an n-dimensional linear equation representing the dynamic of the n-link snake robot. In the direct dynamic formulation, inputs are the n-dimensional vector of joint angles $[\varphi_1, \varphi_2, \ldots, \varphi_{n-1}]$ and the outputs are n unknown variables $\ddot{\theta}_1 \in \mathrm{R}$ and torque, $T \in \mathrm{R}^{n-1}$. Therefore, by solving Eq. 19, we can obtain the joint torques, T_i, and tail link rotation acceleration, $\ddot{\theta}_1$. Substituting these values back into Eq. 18 will obtain acceleration of the tail end, $\ddot{r}_b$. The tail link joint angle (θ_1) and its angular velocity ($\dot{\theta}_1$) as well as velocity of the tail end ($\dot{r}_b$) and its moving distance (r_b) are all obtained through integration. The complete parameters defining robot motion are derived for the case when changes in body shape, φ, are known. Therefore, upon specifying changes in body shape, we can drive at necessary joint torques to generate the desired robot motion. Our use of the derived dynamic equation in this paper is twofold: (1) as simulation tool for evaluation of the proposed control schemes (2) for calculation of fitness function of GA in Sect. 3.2.

3 CPG control locomotion

The CPGs found in vertebrates are composed of neural oscillators. Real neurons have very complicated behaviors and, therefore, it is difficult to build a mathematical model to closely simulate it. CPG-based approaches for locomotion control often use systems of coupled nonlinear oscillators for generating the traveling waves necessary for locomotion ([10,14,15]). These approaches are implemented as differential equations integrated over time with the goal to produce the traveling wave as a limit cycle. The system is then robust against transient perturbation as it asymptotically returns to limit cycle. In this article, we use CPGs model which is originally proposed by Crespi et al. [15] to generate trajectories of relative joint angles for snake robot. Limit cycle of the CPGs has a closed form solution with explicit frequency, amplitude and wavelength. The CPGs is composed of double chain of oscillators with nearest neighbor coupling (Fig. 2).

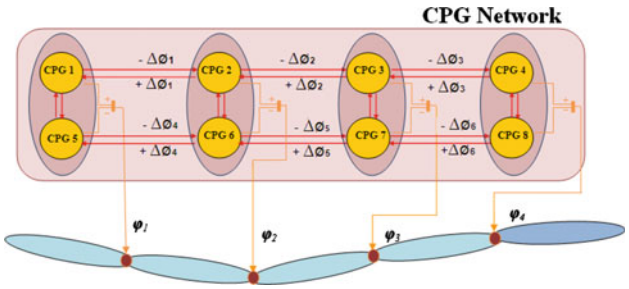

Fig. 2 CPG network for snake robot

The CPG is implemented as the following system of $2N$ coupled oscillators [15]:

$$ \begin{aligned} \dot{\theta}_i &= 2\pi\nu_i + \sum_j w_{ij}\sin(\theta_j - \theta_i - \phi_{ij}) \\ \ddot{r}_i &= a_i\left(\frac{a_i}{4}(R_i - r_i) - \dot{r}_i\right) \\ x_i &= r_i(1 + \cos(\theta_i)), \end{aligned} \quad (20) $$

where the state variables θ_i and r_i are phase and amplitude of the ith oscillator, parameters ν_i and R_i determine the intrinsic frequency and amplitude, and a_i is a positive constant. Weights w_{ij} and phase biases ϕ_{ij} are used to define coupling between oscillators. Variable x_i is the rhythmic and positive output signal extracted out of oscillator i. The values of joint angles, φ_i, are determined by taking the difference between signals from the top and bottom oscillators.

$$ \varphi_i = x_i - x_{i+N} \quad (21) $$

To obtain limit cycle of CPG, we need to select values for parameters defined in Eq. 20. The phase biases, $\emptyset_{ij}$, between paired oscillators which are working on each joint are set equal to π. Phase biases between neighbor oscillators i and $i+1$ are set to + $\Delta\emptyset_i$ and those between $i+1$ and i are set to - $\Delta\emptyset_i$. The frequency parameters are equal for all oscillators, i.e. $\nu_i = \nu$. We also name all amplitude parameters on top side of the CPGs to be $R_i = T_i$ for $i = (1, \ldots, N)$ and $R_i = B_i$ for the bottom side, $i = (N+1, \ldots, 2N)$. It was proved that for the case of coupling two oscillators with dynamic Eq. 20, CPGs converge to sinusoids with a fixed phase difference [15]. Therefore, in the case of multiple coupled oscillators with different values of phase bias, $\Delta\emptyset_i$, CPGs asymptotically converge to a limit cycle with phase difference that is equal to summation of phase biases between each coupled

Table 1 CPG settings for serpentine gait

Parameter names	Symbol	Value
Top side amplitude parameters	T_i	T
Bottom side amplitude parameters	B_i	B
Connection weights	w_{ij}	4
Positive constant	a_i	10
Phase bias, between paired oscillators	ϕ_{ij}	π
Phase bias, between descending neighbor oscillators	ϕ_{ij}	$-\Delta\phi$
Phase bias, between ascending neighbor oscillators	ϕ_{ij}	$+\Delta\phi$

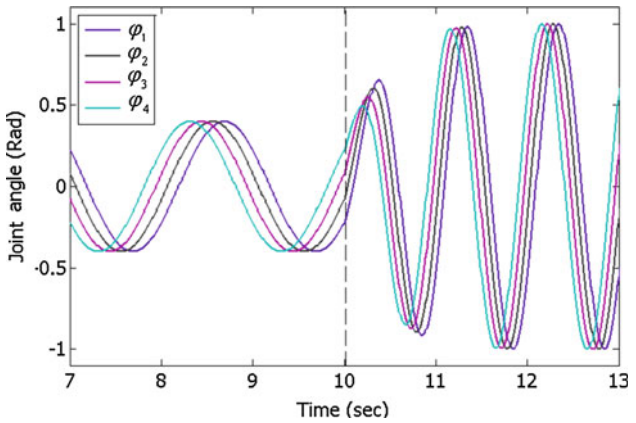

Fig. 3 Oscillatory angles $\varphi_1(t)$ and $\varphi_4(t)$ for serpentine gait ($T = 0.25$, $B = 0.25$ and $\nu = 0.5$ are doubled at $t = 10$ s)

CPGs. The limit cycle is then defined by the following closed form solution for the ith actuated joint:

$$\varphi_i(t) = T_i - B_i + (T_i + B_i)\cos\left(2\pi\nu t + \sum_{j=1}^{i}\Delta\phi_j + \phi_0\right), \tag{22}$$

where $\emptyset_0$ depends on the initial conditions. In this equation, the values of $T_i - B_i$ and $T_i + B_i$ determine offset and amplitude of the ith joint angle, respectively. For the rest of this paper, we name CPGs parameters along with coupling coefficients, CPG network parameters.

3.1 Realization of serpentine gait

Serpentine movement shown in Fig. 4 is one we see in almost all snakes. The most straightforward way to generate serpentine gait in a serial chain is by having the joint angles vary sinusoidally with a common frequency and a constant phase-lag between consecutive joints. Considering Eq. 22, setting CPG parameters as Table 1 leads the robot to move with serpentine gait.

To change robot speed, we use CPG to change the values of amplitude $(T + B)$ and frequency (υ) at t =10 s, while

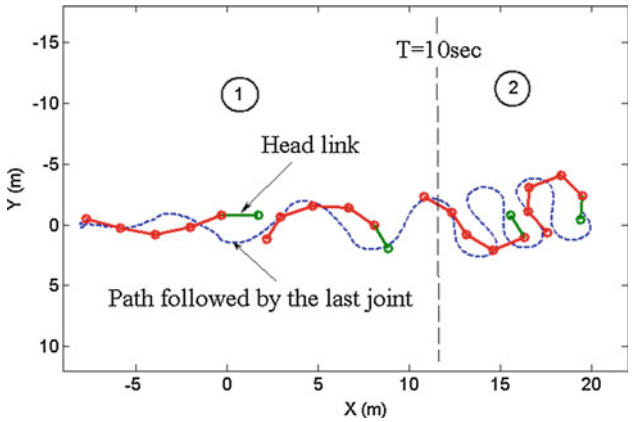

Fig. 4 Snake robot moving with serpentine gait (Motion is the result of relative joint angle shown in Fig. 3)

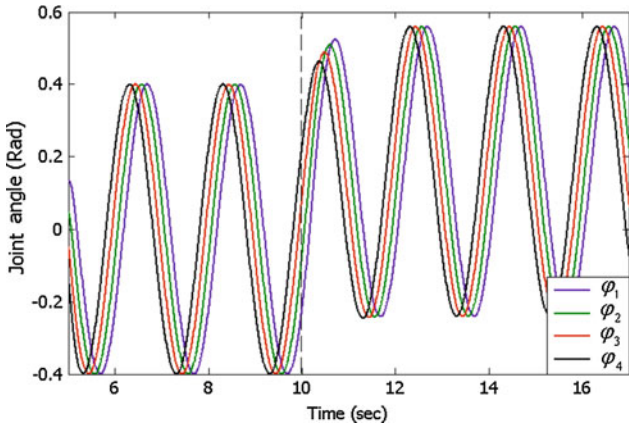

Fig. 5 Oscillatory angles $\varphi_1(t)$ and $\varphi_4(t)$ for serpentine gait (offsets are increased at t=10 s)

other control parameters remain constant (Fig. 3). Resultant motion of the robot is shown in Fig. 4. To obtain this motion, joint angles generated by CPGs, dynamic equation of CPGs (Eq. 20) is solved along with dynamic equation of the robot (Eq. 18 and Eq. 19). Euler method is used for solving both set of equations with the same integration step size, 1ms. Robot parameters used in simulation are l_i=2m, d_i=1m, J_i=0.33 kgm^2, μ_t=0.05, μ_n=0.5. Next, to induce rotation, value of offset $(T - B)$ is changed at t =10 s, while other control parameters remain constant (Fig. 5). The resultant motion of the robot is simulated using dynamic equations and is shown in Fig. 6.

3.2 Realization of spinning gait

In this section, we introduce spinning gait which will allow robot to turn with minimum radius. When the robot encounters an obstacle, it is desirable to turn around as quickest as possible with minimum radius of curvature. Spinning gait further improves snake's maneuverability. In this locomotion

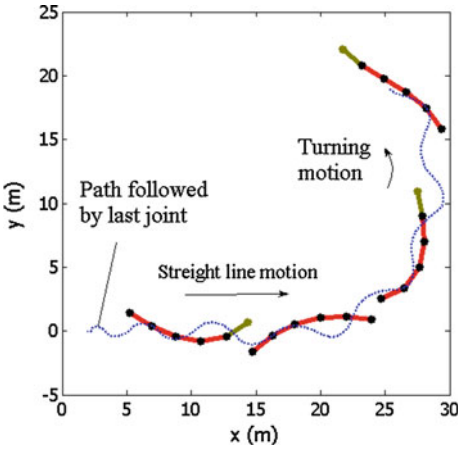

Fig. 6 Snake robot turning with serpentine gait (Motion is the result of relative joint angle shown in Fig. 5)

mode, robot spins around itself while its mass center follows a circle with a small radius. Serpentine gait may also be used to obtain a rotational behavior of snake. This is performed by setting a large value for offset, T-B. However, this locomotion results in entire body of snake following a circular path. This can potentially increase the possibility of collision with obstacles. Therefore, we present a solution on how to adjust CPG network parameter to generate spinning gait.

Problem of finding CPG network parameters for spinning gait is an optimization problem that can be solved with different methods. Considering the total system (combination of mechanical system and neural system), the total dynamics is too complicated to discuss analytically. In such case, synthetic approaches are more effective. Since derivative of the objective function is not available, only derivative-free optimization methods such as Simplex Search or Genetic Algorithms can be used. Although GA is computationally expensive, in this paper, we use it as optimization method, since finding spinning gait parameters is an offline procedure. GA process steps are listed as follows:

- *Optimization parameters* To reflect the symmetries of the robot and to reduce the number of parameters to be optimized, several parameters are set to the same values. Phase biases between opposite neighbor oscillators are set to be equal (i.e. $\Delta\emptyset_i = \Delta\emptyset_{i+N-1}$ for instance $\Delta\emptyset_1 = \Delta\emptyset_4$). Therefore, optimization parameters reduced to 11(T_1–T_4,B_1–B_4, $\Delta\emptyset_1$–$\Delta\emptyset_3$).
- *Constraints* Because of mechanical limit of motor rotation angle, the value for the relative joint angle is also bounded. Considering Eq. 22, the maximum value of the relative joint angle is constrained as,

$$|T_i - B_i| + |T_i + B_i| \leq \varphi_{\max,i} (i = 1, \ldots, N), \qquad (23)$$

where $\varphi_{\max,i}$ is the maximum rotation angle of ith joint.

- *Fitness function* Fitness function in our study is a function of CPG network parameters and is calculated by measuring orientation of the mid-link during the fixed time of locomotion simulation, Eq. 24.

$$\text{Fitness Function} = 1/\Delta\theta_{\text{midlink}} \qquad (24)$$

To find CPG network parameters which generate spinning gait with maximum speed, the difference between minimum and maximum values of the measured mid-link absolute orientation has to be maximized. Here, amount of rotation of the mid-link is used as a quantitative measure for the rotation of the whole robot. Calculation of fitness function is related to mechanical dynamics of the robot as well as friction coefficients of the surface which the snake robot moves on. After adjusting the CPG network parameters, Eq. 20 is solved, and relative angle of adjacent links, φ_i, is calculated. Using these values, dynamic Eq. 19 and then Eq. 18 is solved, and motion of the snake robot is derived (Sect. 2.2). The time of simulation is crucial for achieving good results. Stable locomotion is usually reached after short amount of time. Prior to this time, instable locomotion, we may get a positive fitness score that should be ignored. Therefore, we found, letting the simulation run a certain amount of time before starting the fitness evaluation solves this problem. In this study, we used the orientation of the robot after four simulated seconds as starting configuration and the orientation after 30 s as end configuration to measure the rotation. We also introduce additional constraint for the fitness function. We require that motion of the center of mass of the whole system must remain in a limited area. We define this area as a circle with radius of 80 % of the length of the robot. If robot exceeds the circle during simulation, corresponding genome is destroyed. If this constraint is not introduced, the optimized parameters found by GA will be that of a serpentine motion.

We find CPG network parameters for given $\upsilon = 0.5$ rad/s and $\varphi_{\max} = \pi/2$. Environmental conditions is parameterized by parameters μ_t=0.05 μ_n=0.56. After 300 generations, best fitness value converges to 0.0708. Best individuals corresponding to the best fitness value are $T_1 = 0.445$, $T_2 = 0.445$, $T_3 = 0.785$, $T_4 = 0.785$, $B_1 = 0.785$, $B_2 = 0.785$, $B_3 = 0.445$, $B_4 = 0.445$, $\Delta\emptyset_1 = -0.284$, $\Delta\emptyset_2 = 0$, $\Delta\emptyset_3 = -0.284$.

Interestingly, optimal CPG parameters have the following properties: phase differences of the generated waves tend to be equal except the second one that tends to be zero. Amplitudes of the generated waves tend to be equal. Offset terms also converge to approximately the same values with opposite signs for first and second CPG relative to third and fourth one. Using these parameters and the dynamic equations, snake robot achieves spinning gait shown in Fig. 7. The center of mass of our 5 m long snake moves with speed of 27.16 deg/s in a circle with radius of 3.5 m. As shown, snake robot realizes a sharp turning using spinning gait. The relative joint

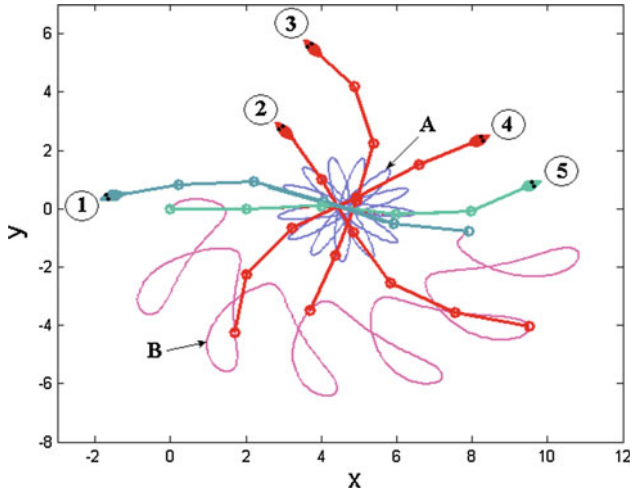

Fig. 7 Snake robot moving with spinning gait (**a** path followed by center of mass of the robot, **b** the path followed by tail of the robot. Poses are numbered through time)

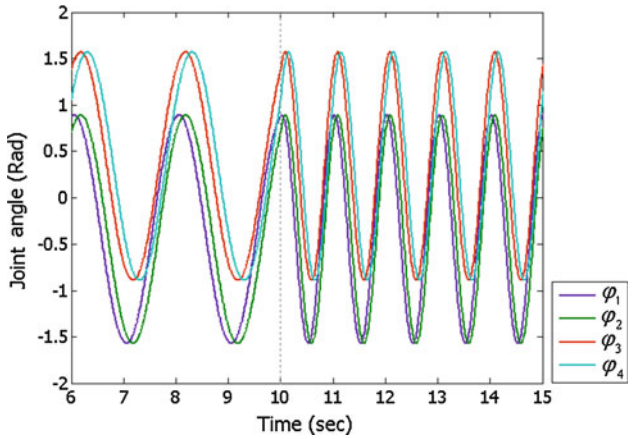

Fig. 8 Relative joint angles $\phi_1(t) \sim \phi_4(t)$ for spinning gait, while frequency is doubled at $t = 10$ s

angle trajectories generated by optimized CPG network are shown in Fig. 8.

As demonstrated in this section, we optimized a specific criterion of the robot motion while giving a general periodic pattern to the robot joint angles in the form of sine waves. The result was interestingly a new locomotion mode for the robot. Other criteria of the robot motion might be defined and optimized to produce an improved motion in terms of energy efficiency or ease of control. For instance, the authors used the same procedure to find a gait called FHS gait in which the aim was to keep the snake robot head toward target during motion [5]. Therefore, this procedure might be used as a general frame work for finding new gaits for robots which propel themselves by periodic undulations of their bodies.

3.2.1 Sudden changes in spinning CPG parameters

Often time, during motion, we may need to change the locomotion parameters for example to change the speed of motion. In this section, we demonstrate that CPG-based controller allows us to change parameters of spinning gait so that joint angles can change in a continuous and rather smooth manner. If a simple sine function was used to generate the motion then online modifications of the parameters of the sine function (e.g. the amplitude or the frequency) will lead to discontinuous jumps of joint angles. This will result in instantaneous torque changes and thus jerky movements.

Regarding to Eq. 22, spinning gait speed can be changed by means of changing frequency (υ). This change is performed at time equal to 10 s, while all other parameters remain constant. Results are shown in Fig. 8. As shown, CPG-based controller can generate continuous and rather smooth trajectories for spinning gait in spite of abrupt change in frequency, doubling of rotational speed.

4 Smooth transition between locomoion modes

Many animals are capable of using multiple gaits. Often simple electrical stimulation of a particular region of the brain stem in vertebrate animals can induce dramatic gait changes, for instance from walk to trot to gallop in cat. Several computational studies of gait transitions exist, for instance by modeling gait transitions as bifurcation phenomena but few have been applied to robotics. In snake robot, the ability to use different modes of locomotion increases maneuverability of the robot in an environment full of obstacles such as search and rescue operations. Therefore, constructing a control structure that utilizes different locomotion modes is an issue of great importance. This control structure should offer the capability of continuous and smooth transition between locomotion modes when it receives command from higher level controller to change the locomotion gait.

Since differential equations of CPGs typically act as first or second order filters, they typically produce smooth modulations of the produced trajectories even when the control parameters are abruptly changed. This feature of CPGs has been used by researchers to produce smooth changes in speed or direction of robots which moves with a specific gait [14]. However, to the best of authors' knowledge, the transition point when a complete gait change occurs has not been studied in previous researches. There are instances when we need to abruptly change a whole set of CPG parameters to a new one. In this section, our aim is to verify by simulation that CPG-based trajectory generation is indeed an ideal solution for this problem. Further, in the paper, we

Table 2 CPG network parameters for spinning and serpentine gaits

	Parameter	Serpentine	Spinning
Top side amplitude parameters	T_1, T_2	0.445	0.445
Top side amplitude parameters	T_3, T_4	0.445	0.785
Bottom side amplitude parameters	B_1, B_2	0.785	0.785
Bottom side amplitude parameters	B_3, B_4	0.785	0.445
Connection weights	w_{ij}	4	4
Positive constant	a_i	10	10
Phase bias, between paired oscillators	ϕ_{ij}	Π	Π
Phase bias, between neighboring oscillators	$\Delta\phi 1$, $\Delta\phi 3$	0.284	-0.284
Phase bias, between neighboring oscillators	$\Delta\phi 2$	0.284	0

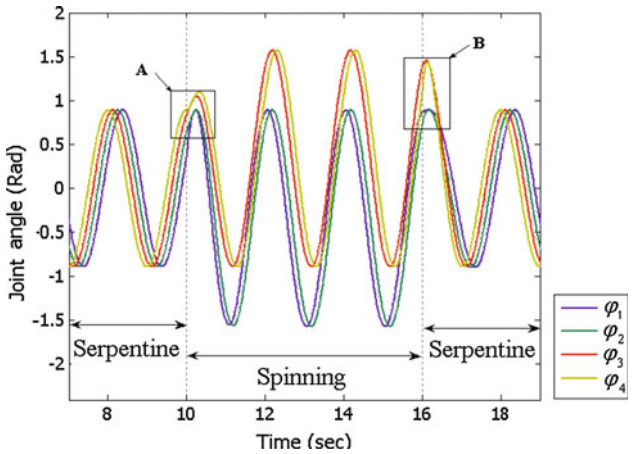

Fig. 9 Oscillatory angles $\varphi_1(t) \sim \varphi_4(t)$ of snake robot while robot transits from serpentine gait to spinning gait and vice versa

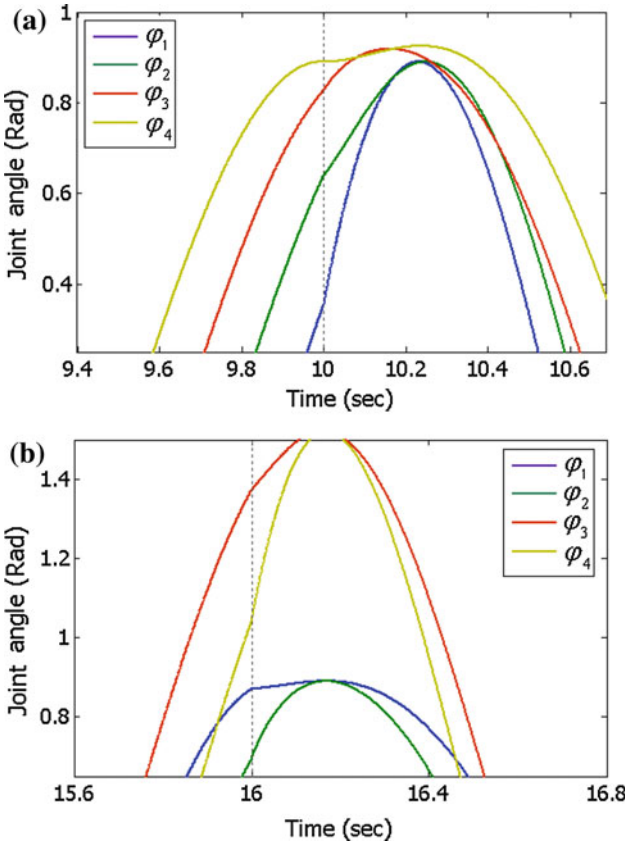

Fig. 10 Up view of area **a** and **b** shown in Fig. 9

will demonstrate this experimentally using constructed snake robot.

In Sect. 3.2, we introduced spinning gait. We also defined a set of CPG network parameters that generate spinning as well as serpentine gaits. We now show by changing CPG network parameters from serpentine to spinning and vise versa, smooth transition will be obtained. To demonstrate this capability by simulation, we will first move in a straight line serpentine locomotion, next use spinning gait to change direction and follow with another straight line serpentine locomotion. We will show that our selected CPG model along with obtained CPG network parameters allows continuous and rather smooth transition between these gaits.

For serpentine gait, we set CPG network parameters as those listed in the third column of Table 2. Spinning gait parameters are those obtained in the previous section (4th column of Table 2). Serpentine gait parameters switch to spinning gait parameters at t =10 s and then switch back to serpentine at t = 16 s. Figure 9 shows relative joint angles φ_1 to φ_4 generated by CPG-based controller, while snake robot changes its locomotion modes. Close up views of joint relative angles at the 10 and 16 s are shown in Fig. 10.

As shown in these figures, relative joint angles change without any discontinuity at the transition times. Furthermore, the transitions occur in a rather smooth manner. Our conclusion for "smooth" transition is simply by close inspection of the slopes of the relative joint angles at the transition times, 10 and 16 s. It is clear that slopes of all four curves are very similar. For these reasons, we can assume

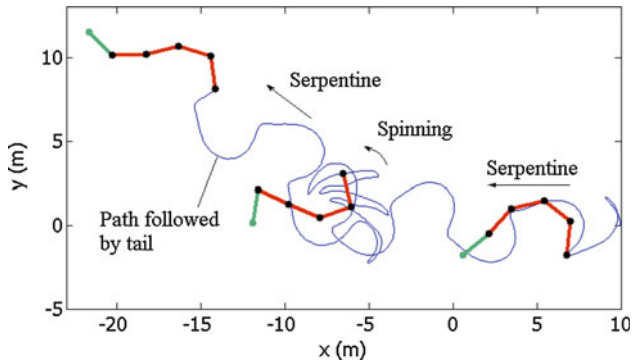

Fig. 11 Path followed by tail of robot while snake moves with multi-modal locomotion

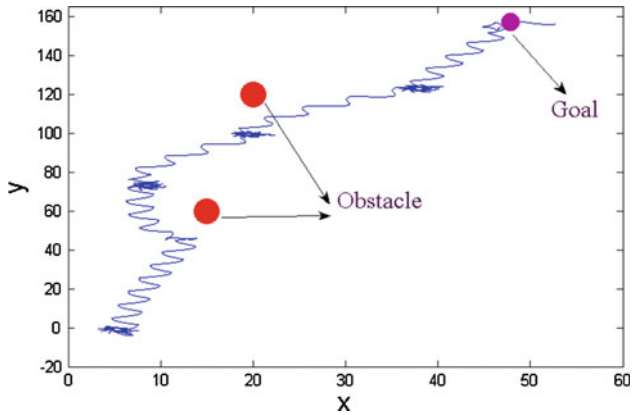

Fig. 12 Path followed by center of mass of the robot during multimodal locomotion

that the relative joint angles change in a rather smooth way. Therefore, we can conclude that, due to continuous and rather smooth transition, the robot does not need to be stopped or reset between iterations and generated trajectories does not lead to jerky movement.

In addition to continuous transition of the relative joint angle trajectories, snake robot also successfully realizes transition between serpentine and spinning gaits. To demonstrate this, simulation is carried out, where dynamic Eqs. 18–19 along with CPG Eq. 20 are solved. Figure 11 shows path followed by tail of the robot during multi-modal locomotion. As illustrated robot can switch from straight-line serpentine gait to straight-line serpentine gait with different direction using spinning gait.

5 Application: locomotion and obstacle avoidance

As an application, consider a goal of guiding a snake robot to its final destination while passing through obstacles. Our snake robot is equipped with proximity sensors mounted in its head link for detecting obstacles. When an obstacle is sensed, locomotion gait changes from serpentine to spinning. The CPG maintains the spinning gait until obstacle is no longer sensed and will smoothly transit to serpentine gait. The proposed multi-modal locomotion, demonstrated in Fig. 12, is an example where snake robot maneuverability is increased.

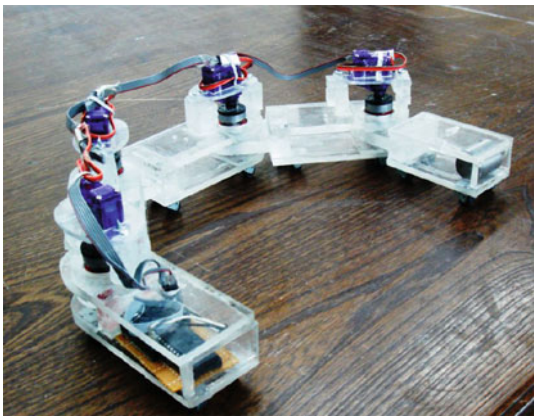

Fig. 13 Experimental model of snake robot

6 Experimental results

In order to experimentally evaluate the multi-modal locomotion presented in previous sections, an undulatory robotic prototype has been developed, using off-the-shelf components and conventional fabrication techniques. The prototype used in the present study, shown in Fig. 13, is composed of 5 Plexiglas links (weight 80g, length 110mm, width 40mm and height 30mm), with the rotary joints actuated by high-torque servo-motors. Each link of the robot is equipped with four wheels which provide differential friction in the tangential and normal directions of motion. The system is powered by on-board batteries or alternatively an external power supply during extended testing sessions. Equation 20 is solved in the microcontroller using Euler method, and results, relative joints angle, are provided as input to each motor.

6.1 Experimental verification of dynamic model

All simulation results obtained thus far have used the derived dynamic equations. In this section, we validate our dynamic model by making our snake robot move in serpentine gait. In order to validate dynamic equation of the robot, Eq. 19, we adjust geometrical parameters (length, mass, link inertia) of the simulated model to represent the physical model. The CPG network parameters of the experimental model are selected the same as those used for simulated model and are shown in the third column of Table 2.

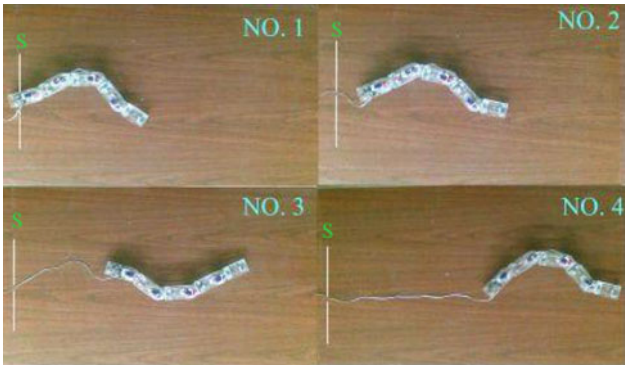

Fig. 14 Snake robot moving with serpentine gait

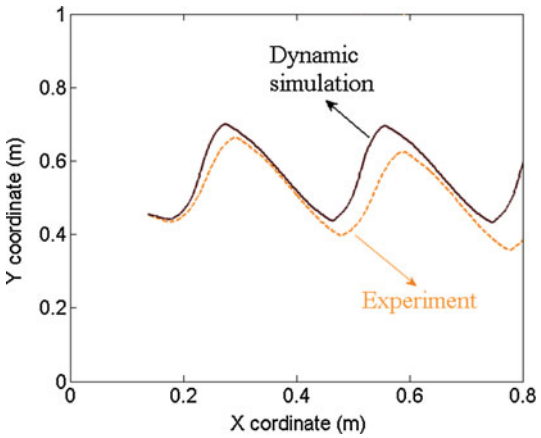

Fig. 15 Comparison of motion predicted by simulation and experimental model

The coefficients of tangential and normal frictions for the actual surface are physically measured and are found to be μ_t=0.05 and μ_n=0.56. Position of center of the mid link is measured by analyzing pictures taken during robot motion. To do this, a digital camera is held fixed overlooking a fixed area. Every 2 s an image is taken. The images are next analyzed off line, and the position of center of the mid link is manually recorded (see Fig. 14).

We compare path followed by center of the mid link of the experimental model (dash line in Fig. 15) with the same path for simulated robot (full line in Fig. 15). Differences between these paths are mainly due to inaccurate friction coefficients and incomplete dynamic equations for ignoring effects, such as joint friction, gearbox and small differences between links. Additionally, the controller used on the physical model is open loop. Therefore, there may be missed encoder counts. Another source of discrepancy between the two results may be due to difficulty in recording the actual path followed by the experimental model. Considering limitations discussed above, it can be concluded that using the dynamic equation a good approximation of the actual motion of the robot can be obtained.

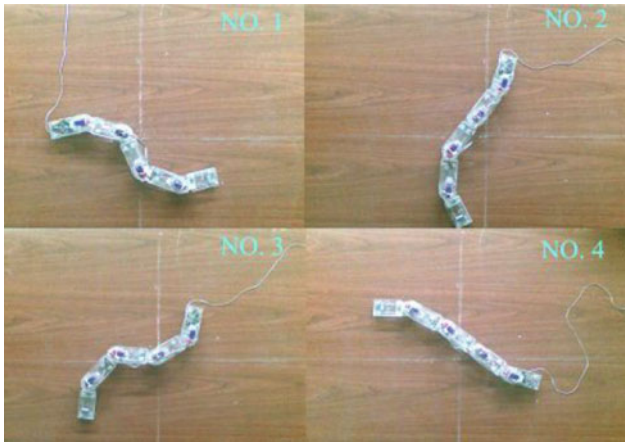

Fig. 16 Snake robot moving with spinning gait

To experimentally generate spinning gait, we adjust CPG parameters as those listed in the 4th column of Table 2. As shown in Fig. 16, robot realizes spinning motion with average speed of 15deg/s.

6.2 Experimental realization of multi-modal locomotion

In this section, we experimentally show that smooth transition between serpentine and spinning gait can be realized using CPG-based controller. Results are shown in Fig. 17. First, CPG network parameters are adjusted to produce serpentine gait (Fig. 17, No. 1–No. 3). Next, we abruptly change to spinning gait parameters settings of Sect. 3.2 (Fig. 17, No. 3–No. 4). Lastly, after a set time, we abruptly change CPG parameters back to serpentine gait (Fig. 17, No. 4–No. 6). Parameters used are listed in Table 2. The resultant motion is a smooth and elegant transition between these two gaits (without any observed stalling or any jerky movement of joints).

The serpentine-to-spinning-to-serpentine locomotion is repeated for different spinning gait times. Path followed by the tail end of the robot is drawn in Fig. 18. Clearly, robot can change its motion to any desirable direction by changing the spinning gait duration time.

7 Conclusion

Ability to moves with different modes of motion offers significant advantage for snake-like robot locomotion and improves its maneuverability. In this paper, we first derived dynamic equation of n-link snake robot using Lagrange's method. We presented a simplified form for the final dynamic equation in matrix format. Next, we introduced spinning gait which allowed the robot to rotate around itself while its center of mass followed a small circle. To do this, CPG net-

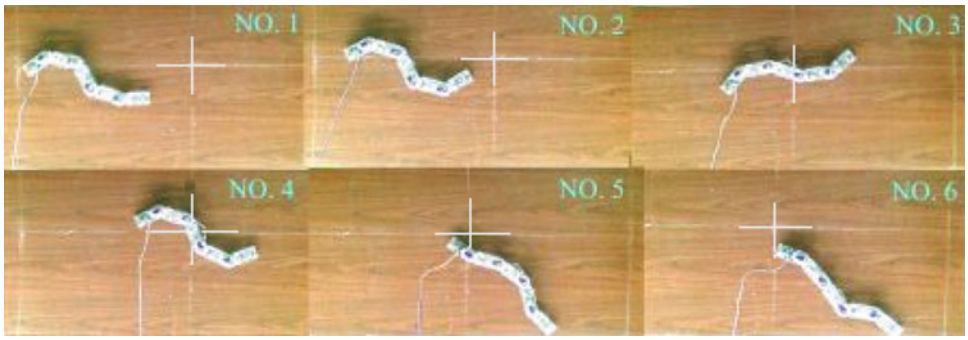

Fig. 17 Snake robot transition between serpentine and spinning gait

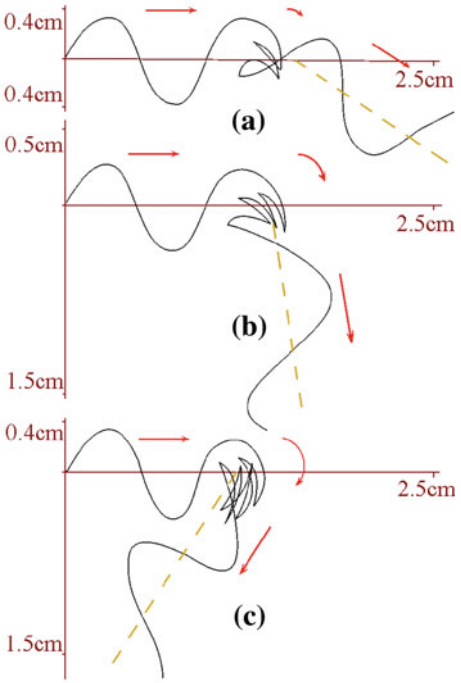

Fig. 18 Robot transition between serpentine and spinning gait for different spinning gait times

work parameters were optimized using genetic algorithm. Using the derived dynamic equations, we illustrated that CPGs allowed the robot to change its gaits naturally like a real snake. Therefore, robot locomotion is closer to its natural counterpart both in terms of motion generation, because of using CPGs as motion generator, and locomotion. An application was presented where CPG controller was used to guide the snake robot from an initial to final position while avoiding obstacles by utilizing serpentine and spinning gaits. We constructed an experimental bed to validate our theoretical results. We verified the dynamics equations by comparing the path of the simulated and physical model. Results showed good agreement indicating correctness of our dynamic model. We experimentally illustrated that CPGs allowed the robot to smoothly change its mode of motion, although a complete gait change requires changing a whole set of CPG parameters. Results presented in this paper can be applied to other robots with undulatory behaviors, such as walking, swimming in which smooth transitions between different trajectories are required. More research needs to be done to draw a general conclusion about capability of CPGs to produce smooth transition between different undulatory behaviors in different robots.

The main contributions of this paper are the introduction of a novel gait called spinning gait, presenting a framework where other gaits may be generated for any robot with undulatory behavior, demonstrating that CPG-based motion generator allows smooth transition between complete gaits and applying it to multimodal locomotion of snake robot.

Acknowledgments The authors would like to thank Ferdowsi University of Mashhad for their financial support in this project

Appendix

The detailed forms of M, H, B and F in Eq. 11 are presented as

$$M_{ij} = \begin{cases} [m_j d_j l_i + (\sum_{k=j+1}^{n} m_k) l_i l_j] \cos(\theta_i - \theta_j) & i < j,\ 1 \le j \le n \\ I_i + m_i d_i^2 + l_i^2 (\sum_{j=i+1}^{n} m_j) & i = j,\ 1 \le j \le n \\ M_{ji} & i > j,\ 1 \le j \le n \end{cases}$$

$$M_{n+1,j} = -\sin\theta_j \left[m_j d_j + \left(\sum_{k=j+1}^{n} m_k \right) l_j \right] (j = 1, \ldots, n)$$

$$M_{n+2,j} = \cos\theta_j \left[m_j d_j + \left(\sum_{k=j+1}^{n} m_k \right) l_j \right] (j = 1, \ldots, n)$$

$$M_{n+1,n+1} = M_{n+2,n+2} = \sum_{i=1}^{n} m_i \qquad (25)$$

$$M_{n+1,n+2} = M_{n+2,n+1} = 0$$

$$H_i = l_i \sum_{j=i}^{n} \left\{ \left[m_j d_j + l_j \left(\sum_{k=j+1}^{n} m_k \right) \right] \sin(\theta_i - \theta_j) \dot{\theta}_j^2 \right\} + \sum_{j=1}^{i-1} \left\{ \left[m_i d_i + l_i \left(\sum_{k=j+1}^{n} m_k \right) \right] l_j \sin(\theta_i - \theta_j) \dot{\theta}_j^2 \right\} \times (i = 1, \ldots, n) \quad (26)$$

$$H_{n+1} = -\sum_{i=1}^{n} \cos\theta_i \left[m_i d_i + \left(\sum_{k=i+1}^{n} m_k \right) l_i \right] \dot{\theta}_i^2$$

$$H_{n+2} = -\sum_{i=1}^{n} \sin\theta_i \left[m_i d_i + \left(\sum_{k=i+1}^{n} m_k \right) l_i \right] \dot{\theta}_i$$

$$B = \begin{bmatrix} D_{n\times n-1} \\ 0 \end{bmatrix}, \quad \text{where } D_{ij} = \begin{cases} -1 & i = j \\ 1 & i = j+1 \\ 0 & \text{others} \end{cases} \quad (27)$$

$$F = \begin{bmatrix} {}^{p}f_{n\times 1} \\ {}^{q}f_{2\times 1} \end{bmatrix} \text{ where}$$

$${}^{p}f_j = d_j (f_{xj} \sin\theta_j - f_{yj} \cos\theta_j) + l_j \left[\sin\theta_j \sum_{i=j+1}^{n} (f_{xi}) - \cos\theta_j \sum_{i=j+1}^{n} (f_{yi}) \right] \quad (28)$$

$${}^{q}f = \begin{bmatrix} -\sum_{i=1}^{n} (f_{xi}) \\ -\sum_{i=1}^{n} (f_{yi}) \end{bmatrix}$$

The detailed final dynamic equation, Eq. 11, has a simplified matrix format and can easily be expanded for any number of links.

References

1. Umetani, Y, Hirose, S (1976) Biomechanical study of active cord mechanism with tactile sensors. In: Proceedings of the international symposium on industrial robots, pp c1-1–c1-10
2. Hopkins, JK, Spranklin BW, Gupta, SK (2009) A survey of snake-inspired robot designs. Bioinspir Biomim 4(2). doi:10.1088/1748-3182/4/2/021001
3. Dowling K (1997) Limbless locomotion, learning to crawl with a snake robot. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1997)
4. Ostrowski J, Burdick J (1996) Gait kinematics for a serpentine robot. In: Proceedings of IEEE international conference on robotics and automation. IEEE, New York, pp 1294–1299
5. Hasanzadeh S, Tootoonchi AA (2010) Ground adaptive and optimized locomotion of snake robot moving with novel gait. Auton Robot 28:457–470
6. Transeth AA, Pettersen KY, Liljeback P (2009) A survey on snake robot modeling and locomotion. Robotica 27:999–1015
7. Hirose S (1993) Biologically inspired robots (snake-like locomotor and manipulator). Oxford University Press, Oxford
8. Matsuno, F, Suenaga, K (2003) Control of redundant 3D snake robot based on kinematic model. In: Proceedings of IEEE international conference on robotics and automation, pp 2061–2066
9. Ijspeert AJ (2008) Central pattern generators for locomotion control in animals and robots: a review. Neural Netw 21(4):642–653
10. Crespi A, Badertscher A, Guignard A, Ijspeert AJ (2005) Amphibot I: an amphibious snake-like robot. Robot Auton Syst 50:163–175
11. Wu X, Ma S (2010) CPG-based control of serpentine locomotion of a snake-like robot. Mechatronics 20:326–334
12. Wu X, Ma S (2010) Adaptive creeping locomotion of a CPG-controlled snake-like robot to environment change. Auton Robot 28:283–294
13. Ryu J, Chong NY, You BJ, Christensen HI (2010) Locomotion of snake-like robots using adaptive neural oscillators. Intel Serv Robot 3:1–10
14. Ijspeert AJ, Crespi A, Ryczko D, Cabelguen JM (2007) From swimming to walking with a salamander robot driven by a spinal cord model. Science 315(5817):1416–1420
15. Crespi A, Ijspeert AJ (2008) Online optimization of swimming and crawling in an amphibious snake robot. IEEE Trans Robot 24(1):75–87