CPG-based Control of Smooth Transition for Body Shape and Locomotion Speed of a Snake-like Robot

Zhenshan Bing[1], Long Cheng[1], Kai Huang[2], Mingchuan Zhou[1], and Alois Knoll[1]

Abstract— **In this paper, a lightweight central pattern generator(CPG) model is designed for a snake-like robot, to achieve smooth transition of body shape and locomotion speed. First, based on the convergence behavior of the gradient system, a lightweight CPG model with fast computing time is designed and compared with other widely adopted CPG models. Then, the body shape and locomotion speed transitions in *rolling gait* are simulated based on the proposed CPG model. Compared with the sinusoid-based method, a smooth transition process can be achieved, without generating undesired movement or abnormal torque. Finally, extensive prototype experiments are conducted to demonstrate that the CPG-based control can effectively ensure smooth transition process and avoid abnormal torque, when the body shape and locomotion speed are changed.**

I. Introduction

To meet the growing need for robotic mobility occasions such as disaster rescuing, factory pipes maintenance, and terrorism surveillance, a considerable number of snake-like robots have been developed in the past decades for ground locomotion [1, 2, 3]. Early versions of snake-like robots are equipped with passive wheels, which can achieve stable and fast planar locomotion by swinging their body. Such robots however lack the ability of moving in varied topography [2, 4]. Therefore, more attention has been focused on snake-like robots with 3D locomotion ability. Snake-like robot with lateral and dorsal connected modules can achieve 3D locomotion by changing the shape of the body, which can adapt to different kinds of terrain [3, 5].

Although the locomotion control of snake-like robots has been widely investigated, many vital problems still exist. Smooth gait transition is one of the unsolved problems, which can cause undesired locomotion [6, 7]. The reason is that the transition process requires another sinusoid wave which may have different phase, amplitude, or frequency. Such changes on the sinusoid wave will lead to discontinuous commands for the joint control. This discontinuity will not only cause undesired movement of the joints, but also generate abnormal high torque which may increase the risk of damaging the motor and the gear of the robot. Therefore, it is desired to control the gait transition smoothly without generating undesired movements and abnormal torque.

CPG-based control method is a promising solution for smooth gait transition. CPG network can generate smoothly self-adjusted rhythmic signals for cyclic motions, particularly suitable for snake-like robots with redundant degrees of freedom. Although several CPG-based methods have been adopted to control the snake-like robots [7, 8, 9], the effectiveness of such methods for smoothing the gait transition of snake-like robots with 3D locomotion ability is nevertheless not fully studied yet. The reason is multi-fold. First of all, 3D locomotion is more complex than 2D in the plane, because it requires two waves in the lateral and dorsal plane to drive the robot. This complexity leads to unstable gait transition of the robot. To cope with this complexity and reduce undesired movements, the CPG control strategy needs to combine the two waves together with the phase difference. Secondly, the CPG needs time to compute the operating patterns, usually on MCU (micro control unit) with limited computing power. The length of this computing time will directly affect the control performance. For snake-like robots with redundant degrees of freedom, the computing load is more demanding, as more degrees of freedom means more CPGs to be calculated on the microprocessor with limited computing power. Hence, the design of a CPG controller with the best trade-off between complexity and simplicity is challenging.

To investigate a smooth gait transition of 3D snake-like robots, this paper studies CPG-based control strategy. In particular, we focus on the transition for the changes of body shape and locomotion speed within the same gait mode. In terms of the smoothness, the robot trajectory and the output torque are targeted, which can reflect the locomotion accuracy and the energy consumption. The contributions of this work are summarized as follows.

- Based on the gradient theory, a lightweight CPG model with fast computing time is designed to control the 3D locomotion of a snake-like robot. Our CPG model is at least 3 times faster than the other widely adopted CPG models in the literature.
- Extensive simulations and prototype measurements are conducted on the robot trajectory and output torque for the change of both body shape and locomotion speed. The results prove that the CPG-based method can effectively avoid undesirable movement and abnormal torque.

The rest of this paper is organized as follows: Section II briefly presents related work. The mechanical and electronic hardware of our snake-like robot are introduced in Section III. Then the CPG mathematical model and the signals' synchronization are deducted and analyzed in Section IV. In Section V and VI, a series of simulations and experiments are conducted to prove the effectiveness of our proposed architecture. Section VII concludes this paper with the discussion

Authors Affiliation: [1] Technische Universität München, Fakultät für Informatik. [2] Sun Yat-Sen University.

Email: {bing,chengl,huangk,zhoum,knoll}@in.tum.de

and the presentation of future work.

II. Related Work

To generate the serpenoid curve signals for snake-like robot, the locomotion control architectures can be classified into two types: sinusoid-based and CPG-based. The most widely adopted control strategy is sinusoid-based method [10]. The sinusoid-based method[1] is achieved to control the snake-like robot, by observing the morphology of real snakes, imitating the serpenoid curves with simple sine-like signals. It was firstly presented by Hirose [11] as *serpenoid curve*.

Through abundant experiments, the sinusoid-based method has been proven simplicity for imitating real snakes locomotion and generating gaits of snake-like robots. For those snake-like robots with 3D locomotion ability, Choset [12] proposed a parameterized gait equation by simplifying the serpenoid curve into sine functions, which can achieve a variety of gaits for different application scenarios. Based on the gait equation, Melo [13] conducted numerous experiments to select appropriate parameters for repeatable gaits under mechanical constraints and theoretical rules of modular snake robots, and proposed several indoor and outdoor gaits for a modular snake robot [13].

One major strength of the sinusoid-based approach is the simplicity, because the important parameters that influence the gaits are predefined, like phase, frequency, and amplitude. However, sinusoid-based method inherently depends on time, which may cause undesirable movements during gait transition. The reason is that the transition process requires a change of parameters of control signals, including the amplitude, frequency, and phase difference. Sinusoid-based method will generate discontinuous waves when the transition happens.

Researchers attempt to use CPG-based method to smooth the signal curves during the gait transition. Because CPG can smoothly adjust the rhythmic signals, without the sudden change of set-up points in sinusoid-based method. Crespi [14] built anguilliform swimming salamander robots to achieve the switch of swimming and crawling gaits based on CPG controller. Seo and Slotine[15] developed an open-loop CPG models for a turtle-like robot while successfully generating reference trajectories for fin motions. Ma [4] used a Cyclic Inhibitory based CPG network with feedback to control the serpentine locomotion of snake-like robot. Gomez [5] also developed a modular robot of pitch-yaw-connecting modules by CPG-based method to simplify the locomotion control. Although many undulatory and anguilliform robotics adopt CPG to control the locomotion, few of them are with 3D locomotion ability. Besides, the undesired locomotion may still be introduced during gait transition process, due to the change of the gait properties. And this phenomenon is rarely mentioned, let alone analyzed in detail.

Ma [7] developed an activation function in CPG model to control the body shape and validated the effectiveness by simulating the joints' torque. The drawback is lack of supporting prototype experiments and their snake-like robot only moves in 2D plane. Dorge [6] stated briefly that smooth transition may be not enough to ensure desired movement, and the principle should maintain the gait properties during transition. Their controller propagated the second gait down the snake-like robot during the gait transition process. However, in their paper, the assessment criteria for effective and repeatable transition gaits or supported prototype experiments are not fully presented.

Some typical nonlinear oscillator models [16] used for CPG in snake-like robots implementation are listed as follow: Matsuoka model, Cyclic Inhibitory model [17], Kuramoto Oscillator model, Hopf Oscillator model [8], [15] and Van Der Pol oscillator. However, the online execution time of those mentioned CPG models are rarely investigated.

Inspired by Dorge's viewpoint [6], we divide that gait transition into three aspects: the change of body shape, locomotion speed and gait mode. They are mainly related to the amplitude, frequency and phase difference in sinusoid-based gait equation [12]. Afterwards, a lightweight CPG model is designed and adopted to demonstrate that CPG-based locomotion control can effectively ensure smooth gait transition, when changing the body shape and locomotion speed.

III. Mechanical Overview

Our snake-like robot has a modular design, consisting 13 actuated modules and a head module, as seen in Fig. 1. Those modules communicate with each other via I^{2}C bus. All the output shafts are alternately aligned with the robot's lateral and dorsal planes to generate 3D locomotion. Each module is connected to the adjacent modules and allows a full 180° rotation.

Each module contains a servo and a set of gears to actuate the joint. The DC servo (DS1509MG) has a maximum torque of $12.8 Kg \cdot cm$ and drives a gearbox with a reduction factor of 3.71. For the electronic hardware part, each module has an Arduino Nano board, a printed circuit board and an angle sensor. The Arduino Nano board runs three tasks: controlling the servo, reading the joint angle, and communicating with the other modules. Table I summarizes the technical specifications of our snake robot.

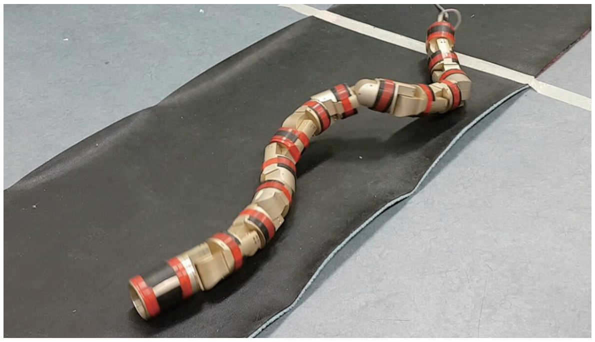


Fig. 1: The snake-like robot with a module design. Each joint axis is orthogonal to its neighbors with a rotation range of ±90°.

TABLE I: Overview of Snake-like robot specifications

Items	Discriptions
Dimensions	Diameter 60mm Length 70cm
Mass	Module 0.3kg Full 2kg
Actuation	Max Torque 12.8kg.cm Max Speed 0.07sec/60°
Power	7.4V DC
Communication	I^{2}C Bus
Sensing	Angular Sensor MLX90316KDC

IV. CPG MODEL BASED ON GRADIENT METHOD

Chain-type CPG network is one of the most common topological structures. The structure of the CPG forms a chain of oscillators coupling the neighbor oscillators.

Inspired by Kopell [18], we design an oscillator model for the chain-coupled CPG based on the convergence behavior of gradient system, which can adjust the output signal's phase difference as desired. More importantly, the model is lightweight with less online execution time.

A. Mathematical model of CPG network

The concept of *Gradient System* comes from the gradient field [19]. For any point M in space region G, there is a certain scalar function $V(M)$, corresponding to point M. Then, $V(M)$ is one certain scalar field in space region G. If there is a gradient function $gradV(M)$, corresponding to the point M, then a gradient field can be determined. It is generated by the scalar field $V(M)$, called the potential of the gradient field. In space region G, the potential of any point decreases against the gradient direction , as shown in(1).

$$\frac{dx}{dt} = -\frac{\partial V}{\partial x}, \quad \vee x \in G \tag{1}$$

If the gradient system has a minimal value x^* in G, then all the vector x will converge to x^*, which satisfies:

$$\left.\frac{\partial V}{\partial x}\right|_{x=x^*} = 0, \quad \left.\frac{\partial^2 V}{\partial x^2}\right|_{x=x^*} > 0 \tag{2}$$

The gradient system has an important property, that any initial state x_0 will converge to the minimal values in region G, during finite time and remain stable since then.

According to (2), we can design a chain-type CPG network to realize a fixed phase difference among all the CPG models. The phase differences among all CPGs can be seen as a group of state vectors. A global convergence gradient system is assumed in G, which is composed by those state vectors. The vectors in G will converge to the extreme point of the gradient system in finite time. If the extreme point of the gradient system is the target phase difference vector as we set, thus the CPG network will converge to the desired phase difference from any position in finite time. As shown in Fig. 2, the chain-type CPG network is composed by n neurons with same parameters. Supposed that, the phase of the i^{th} CPG is $\varphi_i(t)$, and the phase difference between two neighbouring CPGs i^{th}, j^{th} is $\theta(t)$, the desired phase difference decided by the gait generator is $\tilde{\theta}_i$.

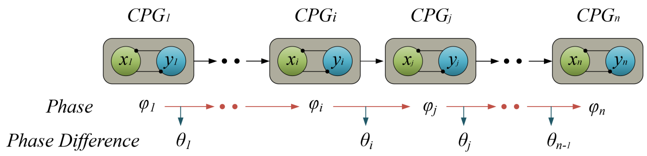


Fig. 2: Parameters setting of CPG net with chain-type.

Consider the chain-type CPG network as a directed graph. The topological structure of the chain-type CPG network can be described by the Incidence Matrix. Set the incidence matrix $T = \{a_{i,j}\}_{(n-1)\times n}$ and satisfies:

$$\{a_{i,j}\} = \begin{cases} -1, & \text{from } j \text{ to } i \\ 1, & \text{from } i \text{ to } j \\ 0, & i \text{ and } j \text{ are not adjacent} \end{cases} \tag{3}$$

Therefore, the incidence matrix T for chain-type CPG network is

$$T = \begin{bmatrix} 1 & -1 & & & 0 \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ 0 & & & 1 & -1 \end{bmatrix}_{(n-1)\times n} . \tag{4}$$

Thus, the relationship between the phase difference and the phase is as follows:

$$\Theta = T\Phi \tag{5}$$

where phase differences vector is $\Theta = [\theta_1, \theta_2, ..., \theta_{n-1}]^T$, the phase vector is $\Phi = [\varphi_1, \varphi_2, ..., \varphi_{n-1}, \varphi_n]^T$.

In order to design the potential function, Ψ_i is used as the generalized coordinates of the gradient system, which satisfies:

$$\Psi_i = \begin{cases} \varphi_1 - \varphi_2 = \theta_1, & i\text{=1} \\ \varphi_{n-1} - \varphi_n = \theta_n, & i\text{=n-1} \\ \varphi_{i+1} + \varphi_{i-1} - 2\varphi_i = \theta_{i-1} - \theta_i, & \text{otherwise} \end{cases} \tag{6}$$

Then the potential function of a parabolic system is as follows:

$$V(\Psi) = \sum_{i=1}^{n} \mu_i(\psi_i - \tilde{\psi}_i)^2 \tag{7}$$

where μ_i is the coefficient of the convergence velocity and $\tilde{\psi}_i$ is the generalized coordinates of the desired phase differences. According to (1), $V(\Psi)$ can be seen as the gradient system. $\tilde{\psi}_i$ is the desired phase difference, represented in the new coordinate constructed by vector Ψ. Thus, the gradient system described by the new coordinate Ψ is:

$$\frac{dV(\Psi)}{dt} = -\frac{\partial V(\psi_1, \psi_2, ..., \psi_n)}{\partial(\psi_1, \psi_2, ..., \psi_n)} \tag{8}$$

Transfer the equation into the original coordinate Θ, we can get:

$$\frac{dV(\theta_i)}{dt} = -\frac{\partial V(\Psi)}{\partial \theta_i} = -\sum_{i=1}^{n} \frac{\partial V(\Psi)}{\partial(\psi_i)} \frac{\partial \psi_i}{\partial \theta_i} \tag{9}$$

Then we can expend (9) into:

$$\frac{d\theta_i}{dt}=\begin{cases}-2\mu_1(\theta_1-\tilde{\theta}_1)-2\mu_2(\theta_1-\theta_2-\tilde{\theta}_1+\tilde{\theta}_2), & i\text{=1}\\ 2\mu_{n-1}(\theta_{n-1}-\theta_n-\tilde{\theta}_{n-1}+\tilde{\theta}_n) & \\ -2\mu_n(\theta_{n-1}-\tilde{\theta}_{n-1}), & i\text{=n-1}\\ 2\mu_{i-1}(\theta_{i-1}-\theta_i-\tilde{\theta}_{i-1}+\tilde{\theta}_i) & \\ -2\mu_i(\theta_i-\theta_{i+1}-\tilde{\theta}_{i-1}+\tilde{\theta}_{i+1}), & \text{otherwise}\end{cases} \tag{10}$$

Finally, the gait generator model for the chain-type CPG-network is obtained as follows:

$$\begin{bmatrix}\dot{\varphi}_1\\ \dot{\varphi}_2\\ \vdots\\ \dot{\varphi}_n\end{bmatrix}=\begin{bmatrix}\omega_1\\ \omega_2\\ \vdots\\ \omega_n\end{bmatrix}+A\begin{bmatrix}\varphi_1\\ \varphi_2\\ \vdots\\ \varphi_n\end{bmatrix}+B\begin{bmatrix}\tilde{\theta}_1\\ \tilde{\theta}_2\\ \vdots\\ \tilde{\theta_{n-1}}\end{bmatrix} \tag{11}$$

where A is,

$$A=\begin{bmatrix}-\mu_1 & \mu_2 & & & 0\\ \mu_2 & -2\mu_2 & \mu_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & -2\mu_{n-1} & \mu_{n-1}\\ 0 & & & \mu_n & -\mu_n\end{bmatrix}_{n\times n} \tag{12}$$

and B is,

$$B=\begin{bmatrix}1 & & & 0\\ -1 & 1 & & \\ & -1 & \ddots & \\ & & \ddots & 1\\ 0 & & & -1\end{bmatrix}_{n\times(n-1)} \tag{13}$$

ω_i in (11) is the integration constants, which is also the frequency of the CPG signal. To output the same frequency signals, we set $\omega_1=\omega_2=...=\omega_n$. The convergence rate of the system is decided by the matrix A, which increases with the value of μ_i.

In summary, combining a PD controller to ensure that the amplitude converges to the desired value [14], the single neuron CPG model can be written as,

$$\begin{cases}\dot{\varphi}_i &=& \omega_i+A\{i,:\}\cdot\Phi+B\{i,:\}\cdot\tilde{\Theta}\\ \ddot{r}_i &=& a_i[\frac{a_i}{4}(R_i-r_i)-\dot{r}_i]\\ x_i &=& r_i\sin(\varphi_i);\end{cases} \tag{14}$$

where $A\{i,:\}$ is the i^{th} row vector in (12), $B\{i,:\}$ is the i^{th} row vector in (13). Φ is the vector of the phase of the CPG neurons and $\tilde{\Theta}$ is the vector of the phase difference amoung the CPG neurons. The state variable φ_i and r_i are the phase and the amplitude of the i^{th} oscillator, respectively. The parameter R_i determines the stable amplitude and a_i is a positive constant. The variable x_i is the rhythmic output signal integrated by the phase ϕ_i and the amplitude r_i. To evaluate the output signals of the proposed CPG model, we plot four CPG outputs by adjusting the parameters with time, as seen in Fig. 3. The CPG network change the phase difference from 0 to π, at $t=10$, then change the frequency from 0.5Hz to 1Hz, at $t=10$, as we desire.

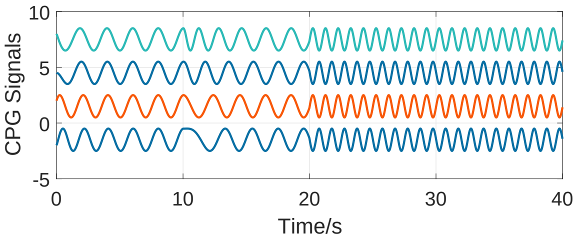


Fig. 3: CPG output waves with changing parameters. Phase difference changes from 0 to π, with θ changing from 0 to π, at $t=10s$. Frequency changes from 0.5Hz to 1Hz, with ω_i changes from π to 2π, at $t=20s$.

To evaluate the convergence speed of the CPG model, we define the convergence tolerance as (15). where n is the number of CPGs, and the target phase difference among the CPG network is 0. Therefore, the signals will reach the peak value at the same pace in finite time. The convergence is treated as finished, when the tolerance is below 1%.

$$tolerance=\sum_{i=1}^{n-1}(\frac{1}{n-1}\mid(x_i-x_j)\mid), \quad j=i+1 \tag{15}$$

The parameter μ in (12), is related to the speed of the synchronization. Based on the tolerance definition in (15). The relationship between the convergence rate and the parameter μ is shown in Fig. 4. Therefore, we can conclude that the convergence rate increases with μ.

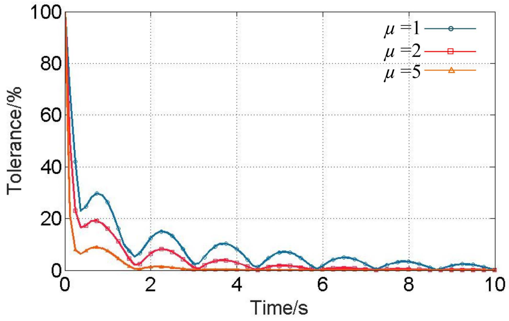


Fig. 4: The synchronization speed of a three-CPG network changes with the parameter μ. Y axis is the tolerance between adjacent CPG.

B. Comparison with other CPG models

Those widely adopted CPG models are based on different nonlinear oscillators, like Kuramoto Oscillator model adopted by Ijspeert and Shugen Ma [14] [7], Hopf Oscillator model adopted by Seo and Slotine [15] and Van Der Pol oscillator [20]. Many of them can generate waves with desired properties, like frequency, amplitude and phase difference. However, the execution time on MCU(Micro Control Unit) is rarely investigated, which is critical factor to influence the performance of the CPG model in real-life implementations. To examine the advantage of the prososed CPG model, experiments are conducted on an Arduino Nano board(Atmel

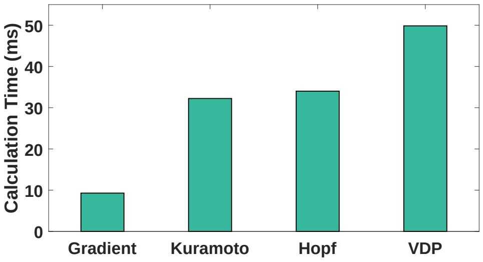


Fig. 5: The execution time on Atmega 328 for 10 iteration of four kinds of CPG models. Each iteration step is 5ms.

Atmega 328) to inspect the online execution of those CPG models. The fourth order Runge-Kutta is adopted to solve the differential equations of a three-CPGs network. To ensure the solution accuracy of coupled differential equations, we set the CPG output period as $50ms$ and the step-size as $5ms$. The online executing experiments results are shown in Fig. 5. We can observe that the online execution time for ten steps of the proposed CPG model is around $10ms$, which can easily satisfy the required signal output period. Since the total online execution involves on-bus data transmit, other CPG models may consume time more that 50 millisecond, which means a potential failure of output period. Therefore, the conclusion can be made that the proposed CPG model is lightweight, compared with other CPG models

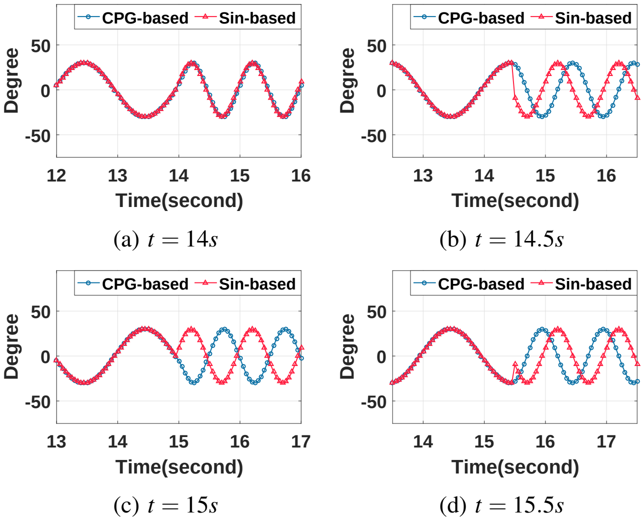
V. Simulations

Two aspects are demonstrated in simulations, compared with sinusoid-based control method. First, the CPG-based control method can ensure smooth gait trajectory, when the body shape and locomotion speed are changed. Second, the CPG-based control method can effectively decrease the abnormal torque during the transition.

A. Gait transition simulations

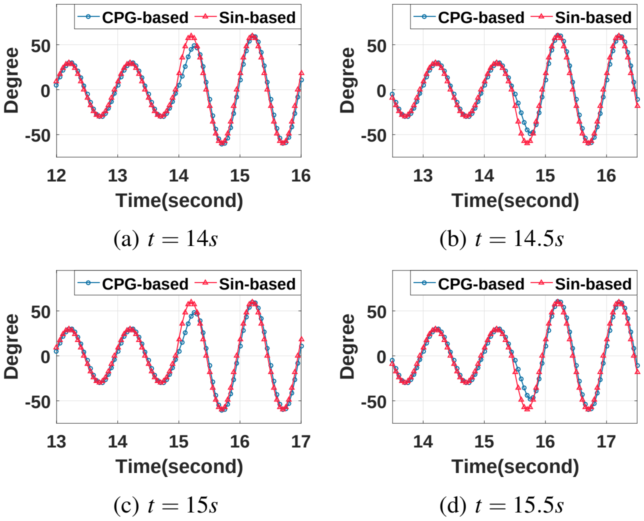
As mentioned before, undesired movement will be generated during unsmooth gait transition process. The reason is that when the robot changes the gait properties, each joint position will change to another sinusoid wave, which probably has different values and direction. Thus, the principle for a smooth transition is to maintain the gait properties during the transition. According to different gait properties, the gait transition can be divided into three kinds, the change of body shape, the locomotion speed, and the gait mode, which are directly related to the amplitude, the frequency and the phase difference of the sinusoid wave.

Besides generating self-adjusted rhythmic signals, CPG can also ensure that the output signals still remain the previous phase difference among all modules. Therefore, if we only adjust the body shape or the locomotion speed by changing the amplitude or the frequency of the signals, the gait mode will not be changed. However, sinusoid-based method can not maintain the phase difference during the transition, due to the fact that inherently relies on time. The compared results between CPG-based method and sinusoid-based method are shown in Fig. 6 and Fig. 7. From Fig. 6(a) to Fig. 6(d), the frequency of the signal is changed from π to 2π, at $t = 14s$, $t = 14.5s$, $t = 15s$ and $t = 15.5s$, respectively. Since sinusoid-based method inherently relies on time, it generates different discontinuous waves at different time. In contrast, CPG-based method exhibits continuously transition process, no matter when the transition happens. From Fig. 7(a) to Fig. 7(d), the amplitude of the signal is changed from 30° to 60°, at $t = 14s$, $t = 14.5s$, $t = 15s$ and $t = 15.5s$, respectively. Although the discontinuity is not obvious as Fig. 6, CPG-based method still exhibits more smooth wave compared to sinusoid-based method. Those time-varying discontinuous waves will cause undesired movement, when all the errors are integrated to the joints of the robot.



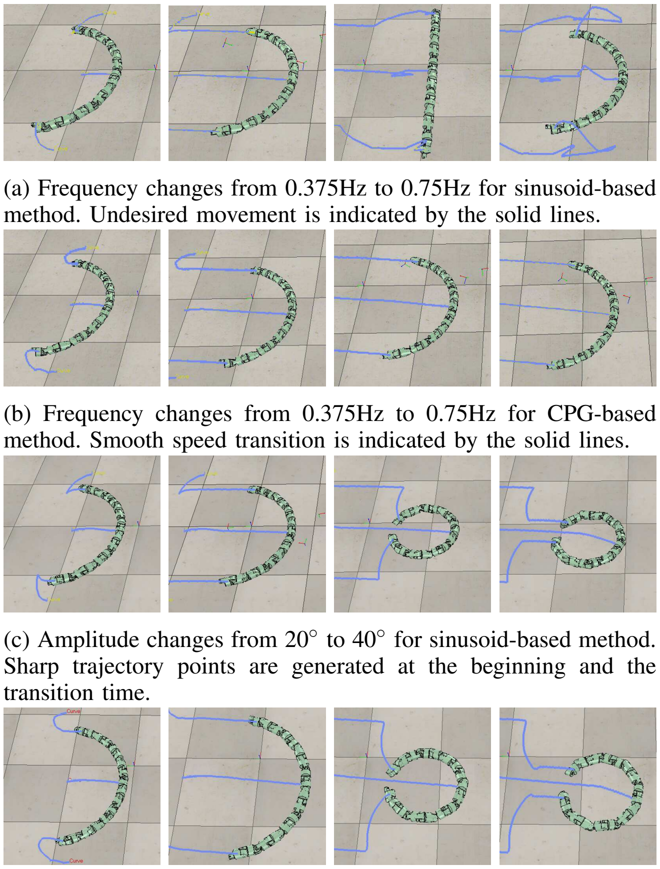
(a) $t = 14s$
(b) $t = 14.5s$
(c) $t = 15s$
(d) $t = 15.5s$

Fig. 6: Change frequency from 1Hz to 2Hz, at four different time in a period.



(a) $t = 14s$
(b) $t = 14.5s$
(c) $t = 15s$
(d) $t = 15.5s$

Fig. 7: Change amplitude from 30° to 60°, at four different time in a period.



(a) Frequency changes from 0.375Hz to 0.75Hz for sinusoid-based method. Undesired movement is indicated by the solid lines.

(b) Frequency changes from 0.375Hz to 0.75Hz for CPG-based method. Smooth speed transition is indicated by the solid lines.

(c) Amplitude changes from 20° to 40° for sinusoid-based method. Sharp trajectory points are generated at the beginning and the transition time.

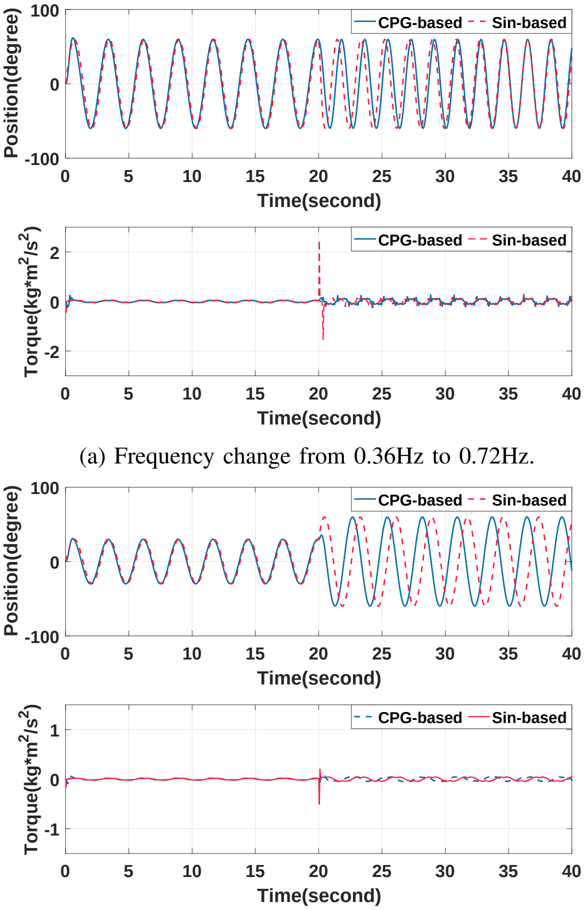
(d) Amplitude changes from 20° to 40° for CPG-based method.

Fig. 8: The body trajectories of the snake-like robot under *rolling* gait, simulated in V-rep. The figures in each row represent the time of the beginning, before transition, transition, and after transition, from left to right.

To examine the trajectory during the gait transition for snake-like robot, a snake-like robot has been developed in Virtual Robot Experimentation Platform (V-Rep EDU Version). Each joint of the robot is actuated by the related signal, generated by sin-based method or CPG-based method. The *rolling gait* [21] is adopted, because it is the one of the most stable and common gaits for snake-like robot. As shown in Fig. 8, three solid lines represent the trajectories of the head module, the center of the body, and the tail module.

The rolling frequency is changed from 0.375Hz to 0.75Hz in Fig. 8(a) and Fig. 8(b), for sinusoid-based method and CPG-based method, respectively. The trajectories in Fig. 8(a) present the undesired movement when the transition happens. In contrast, the trajectories in Fig. 8(b) achieve the locomotion speed transition smoothly.

The rolling amplitude is changed from 20° to 40° in Fig. 8(c) and Fig. 8(d) Although the two figures present desired locomotion, the curves of the trajectories in Fig. 8(d) is more smooth than that in Fig. 8(c). Especially, at the beginning of the trajectories, sharp points are avoided by comparing two methods.



(a) Frequency change from 0.36Hz to 0.72Hz.

(b) Amplitude change from 30° to 60°.

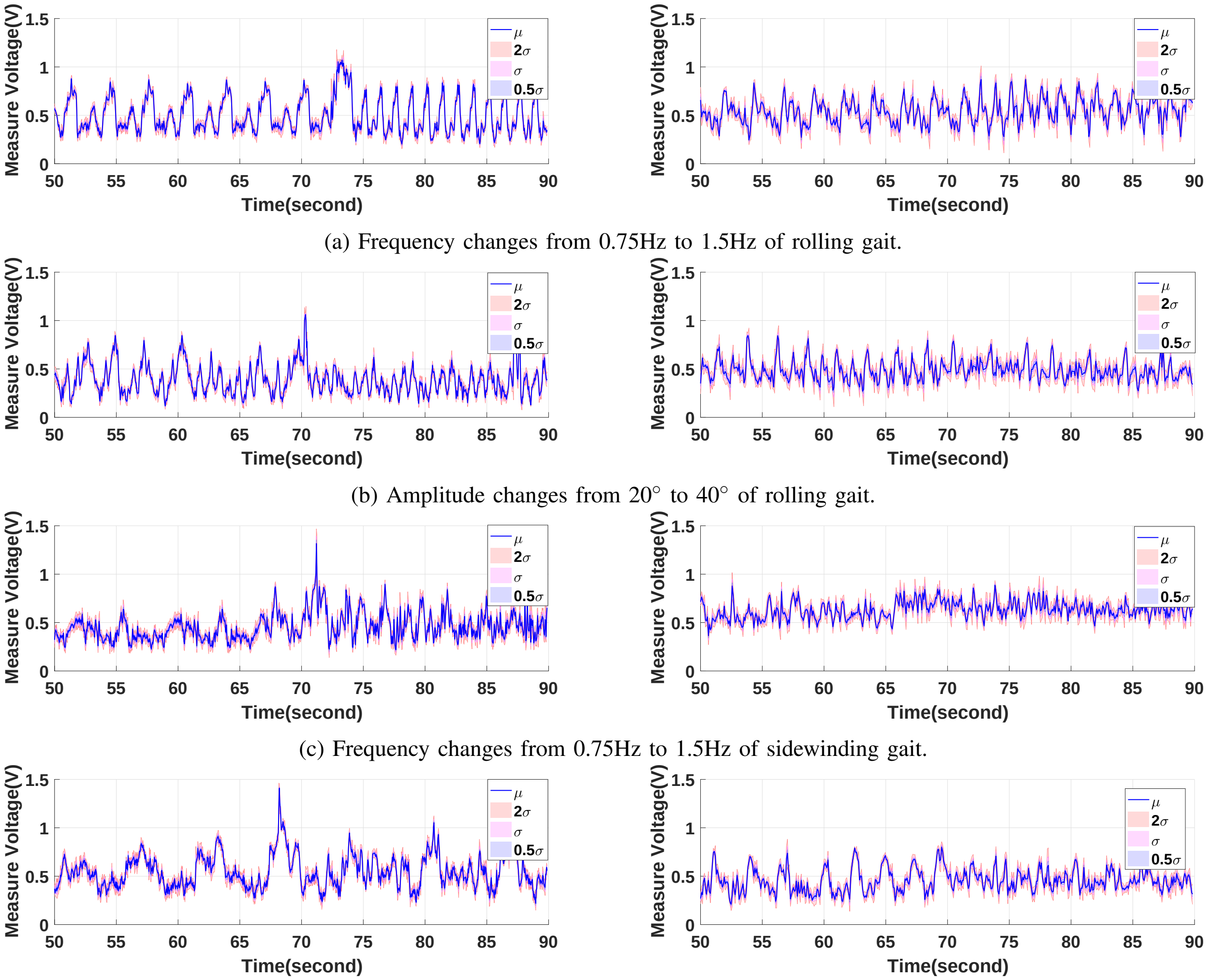
Fig. 9: Torque of one joint controlled by sin-based method and CPG-based method.

B. Torque simulations

Compared to sinusoid-based method, CPG-based method can also decrease the torque during the transition. To compare the output torque, a group of simulations are conducted with two snake modules, connected by one joint. As shown in Fig. 9(a) and Fig. 9(b), the dash lines show that an abnormal high torque is generated, when the transition happens. The abnormal torque is more than 10 times larger than the normal gait torque, when changing the frequency from 0.36Hz to 0.72Hz, in Fig. 9(a). When the body shape is changed from 30° to 30°, a 5 times larger torque than normal is still caused, as seen in Fig. 9(b). However, those CPG-based torque with solid lines have no obvious change point, which can prove that CPG-based method can avoid the abnormal torque when changing the body shape or the travel velocity.

VI. Prototype Experiments

In Section IV, the characters of the proposed CPG model has been proved via numerical simulation and online execution experiments. In Section V, simulations are performed to prove that CPG-based method can achieve smooth transition, when the body shape or the locomotion speed is changed.



(a) Frequency changes from 0.75Hz to 1.5Hz of rolling gait.

(b) Amplitude changes from 20° to 40° of rolling gait.

(c) Frequency changes from 0.75Hz to 1.5Hz of sidewinding gait.

(d) Amplitude changes from 20° to 40° of sidewinding gait.

Fig. 10: Voltage measurements of the high-load resistance. The left column figures are sinusoid-based method measurements. The right column figures are CPG-based method measurements.

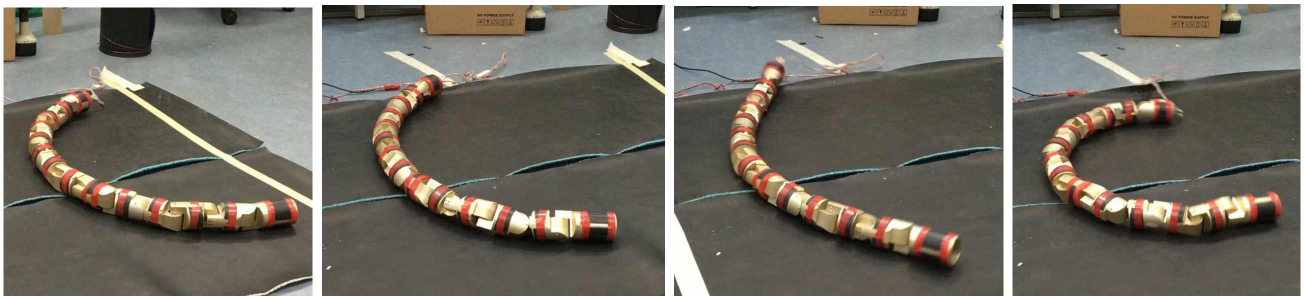


Fig. 11: Screenshots of the amplitude changes from 20° to 40° of rolling gait. From left to right, the figures show the 20° rolling, the state before transition, the jerky movement during transition, and the 40° rolling after transition, respectively. Due to space limit, other frequency changing and sidewinding gait experiments are shown in the attached video.

Now, we report the results of a set of experiments conducted on our snake-like robot to demonstrate that our CPG-based method can ensure smooth transition of the body shape and locomotion speed. The snake-like robot is described in Section III. The CPG network is implemented on a Master Arduino Nano board, which located in the tail module and sends the commands via I^{2}C bus. Meanwhile, it will receive the angle position of each module. Every slave module runs a PD controller to control the servo such that the joint can reach the desired position.

As an illustration of undesired movement, the body shape transition in rolling gait is shown in Fig. 11. The amplitude is changed from 20° to 40°. The four figures from left to right, present the body shape of 20° rolling, the body shape before transition, the jerky movement during transition(the curve of the arc should increase, not decrease), and the body shape of 40° rolling. The undesired movement can not be observed in the contrast experiments, as shown in attached video.

The output torque of the snake-like robot is directly related to the current. Therefore, by measuring the current of the robot, we can obtain the changing trend of the output torque. A $0.1\Omega/50W$ high-load resistance is stringed into the circuit and the voltage of the resistance is measured by oscilloscope. Thus, the changing trend of the torque can be present by the voltage measurements.

As seen in Fig. 10, the results of sinusoid-based method and CPG-based method are in the left column and the right column, respectively. The average value is presented with solid lines and the standard deviation of each figure is represented with σ . Fig. 10(a) describes the frequency transition of rolling gait, from 0.375Hz to 0.75Hz. The transition of the sinusoid-based method happens at $t \approx 73s$, when an abnormal torque is generated. However, the transition of the CPG-based method does not show obvious abnormal torque. The amplitude transition of rolling gait from 20° to 40° is shown in Fig. 10(b), the similar abnormal torque happens at $t \approx 70s$ in the left. On the contrary, by adopting the CPG-based method, the voltage measurements exhibit relative smooth processes. In order to ensure the accuracy of the experiments, *sidewinding gait* is also conducted, changing the frequency and the amplitude as well. In Fig. 10(c), frequency is adjusted from 0.375Hz to 0.75Hz at $t \approx 72s$. In Fig. 10(d), amplitude is adjusted from 20° to 40° at $t \approx 68s$. Both the figures in the left show the abnormal torque when start the transition process. On the opposite side, the CPG-based method, the voltage measurements exhibit no abnormal torque during the transition.

VII. Conclusions and Future Work

In this paper, we have proposed a lightweight CPG model with fast computing time. Then, a CPG-based locomotion control architecture for snake-like robots has been presented and compared with the sinusoid-based method. Simulation and experiment results show that the snake-like robot can achieve smooth body shape and locomotion speed transition by changing the amplitude and the frequency.

For future work, we plan to adopt the decentralized transition [6], for example, transiting the second the gait from the head to the tail order. This could be a potential way to ensure a repeatable locomotion during the gait transition process.

References

[1] S. Hirose, “Biologically inspired robots: Snake-like locomotors and manipulators,” 1987.

[2] M. Mori and S. Hirose, “Three-dimensional serpentine motion and lateral rolling by active cord mechanism acm-r3,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, vol. 1, 2002, pp. 829–834 vol.1.

[3] D. Rollinson, S. Ford, B. Brown, and H. Choset, “Design and modeling of a series elastic element for snake robots,” in *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013.

[4] X. Wu and S. Ma, “Cpg-based control of serpentine locomotion of a snake-like robot,” *Mechatronics*, vol. 20, no. 2, pp. 326–334, 2010.

[5] J. González Gómez, H. Zhang, E. I. Boemo, and J. Zhang, “Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules,” in *International Conference on Climbing and Walking Robots*, 2006.

[6] G. Droge and M. Egerstedt, “Optimal decentralized gait transitions for snake robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 317–322.

[7] N. M. Nor and S. Ma, “Smooth transition for cpg-based body shape control of a snake-like robot,” *Bioinspiration & biomimetics*, vol. 9, no. 1, p. 016003, 2013.

[8] Z. Wang, Q. Gao, and H. Zhao, “Cpg-inspired locomotion control for a snake robot basing on nonlinear oscillators,” *Journal of Intelligent & Robotic Systems*, pp. 1–19, 2016.

[9] M. Sfakiotakis and D. P. Tsakiris, “Neuromuscular control of reactive behaviors for undulatory robots,” *Neurocomputing*, vol. 70, no. 10, pp. 1907–1913, 2007.

[10] P. Liljebäck, K. Y. Pettersen, O. Stavdahl, and J. T. Gravdahl, *Snake robots: modelling, mechatronics, and control*. Springer Science & Business Media, 2012.

[11] H. Ohno and S. Hirose, “Design of slim slime robot and its gait of locomotion,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, vol. 2, 2001, pp. 707–715.

[12] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, “Parameterized and scripted gaits for modular snake robots,” *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, 2009.

[13] K. Melo, L. Paez, and C. Parra, “Indoor and outdoor parametrized gait execution with modular snake robots,” in *2012 IEEE International Conference on Robotics and Automation*, 2012.

[14] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, “From swimming to walking with a salamander robot driven by a spinal cord model,” *science*, vol. 315, no. 5817, pp. 1416–1420, 2007.

[15] K. Seo, S.-J. Chung, and J.-J. E. Slotine, “Cpg-based control of a turtle-like underwater vehicle,” *Autonomous Robots*, vol. 28, no. 3, pp. 247–269, 2010.

[16] J. Yu, M. Tan, J. Chen, and J. Zhang, “A survey on cpg-inspired control models and system implementation,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 3, pp. 441–456, 2014.

[17] X. Wu and S. Ma, “Cpg-based control of serpentine locomotion of a snake-like robot,” *Mechatronics*, vol. 20, no. 2, pp. 326–334, 2010.

[18] N. Kopell and G. B. Ermentrout, “Coupled oscillators and the design of central pattern generators,” *Mathematical biosciences*, vol. 90, no. 1, pp. 87–109, 1988.

[19] B. A. Pearlmutter, “Gradient calculations for dynamic recurrent neural networks: A survey,” *IEEE Transactions on Neural networks*, vol. 6, no. 5, pp. 1212–1228, 1995.

[20] H. Yu, W. Guo, J. Deng, M. Li, and H. Cai, “A cpg-based locomotion control architecture for hexapod robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5615–5621.

[21] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, “Parameterized and scripted gaits for modular snake robots,” *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, 2009.