CPG-based Neural Controller for Serpentine Locomotion of a Snake-like Robot

by

Xiaodong Wu

M.S. (China University Of Petroleum) 2008

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in

Department of Robotics

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of the

RITSUMEIKAN UNIVERSITY, BIWAKO-KUSATSU CAMPUS

Committee in charge:

Professor Shugen Ma, Chair
Professor Sadao Kawamura
Professor Masaaki Makikawa

Fall 2011

CPG-based Neural Controller for Serpentine Locomotion of a Snake-like Robot

by

Xiaodong Wu

Abstract

CPG-based Neural Controller for Serpentine Locomotion
of a Snake-like Robot

by

Xiaodong Wu

Doctor of Philosophy in Department of Robotics

Ritsumeikan University, Biwako-Kusatsu Campus

Professor Shugen Ma, Chair

Even though snakes have elongated and limbless bodies, they can move nimbly on rough ground with various gait patterns. By studying the mechanism of snake motion, these advantageous characteristics can be applied to a snake robot. Bio-inspired snake-like robots are expected to have applications in search and rescue tasks in an unstructured environment, where traditional mobile mechanisms can not gain good access. Previous studies on snake-like robots have mostly focused on the construction of such machines that have these animal-like properties. One of the difficulties in the control of the snake-like robots is how to achieve adaptive locomotion in accordance with environmental information. The highly-redundant structure of the snake-like robot and the large degrees of freedom (DOFs) make the control of the robot complicated.

A considerable number of controllers for snake robots employed the methods that are based on derivation of purposive body motion to design specific controller. Their control architecture can informally be divided into sine based and model based. Little universality and complex modeling have limited these control methods. Recently, a decentralized control method based on the use of simple oscillation generators and on the coordination between them has been attracting a lot of attention. This decentralized control methodology is

inspired by the rhythmic generating mechanism of animals, which is called the central pattern generator (CPG). This kind of mechanism is quite suitable for the control of a snake-like robot with large DOFs, where each joint of the segment has two DOFs. To realize the decentralized control of such robot, CPGs can generate a purposive rhythmic motion pattern flexibly by using the entrainment caused by inter-CPG interaction and the sensory input from the outside environment.

In this study, a bio-inspired CPG-based neural controller is designed for the snake-like robot based on the nonlinear neural model. A novel network with feedback connection is presented which can generate uniform waves with the same amplitude and specific phase difference without additional adjustment. Based on the analysis of the CPG model, the mathematical relationship between the parameters and rhythmic output were investigated. A desired serpentine locomotion of the snake robot can be performed with the parameter modulation of the CPG-based oscillator network. The proposed CPG-based oscillator netowrk can imitate the rhythmic swing of snake joints by use of entrainment caused by inter-CPG interaction and sensory input from the musculoskeletal system. Simulation and physical experiments have shown the effectiveness of the motion control.

By integrating the sensory feedback signals, a self-adaptive motion control method of the snake-like robot has been proposed. The proposed controller was designed as a topological structure composed of two main parts: neural oscillator network (NON) and neural preprocessing network (NPN). The CPG-based oscillator network was constructed as a rhythmic motion generator with nonlinear dynamics. Meanwhile, the neural preprocessing networks can integrate external signals to generate a driving input for CPG network and activate an appropriate reactive behavior. To simplify the model of the kinematics, Head-navigated locomotion and AMM-based turn motion are presented firstly with detailed modeling analysis. With the help of the proposed approach, the control system of the snake-like robot can combine the motion and environmental information, and evolve to an automatically optimal configuration on its own. This provided a flexibility for the integration of multi-input for the neural oscillator network which enabled versatile reactive behaviors to be realized for the snake robot.

TABLE OF CONTENTS

Contents

Contents **i**

List of Figures **v**

List of Tables **ix**

Acknowledgements **x**

1 Introduction **1**

1.1 Research Background 1

1.1.1 Applications 2

1.1.2 Challenges of Limbless Locomotion 4

1.2 Biological Snakes 4

1.2.1 Anatomy of Snakes 5

1.2.2 Locomotion of Snakes 6

1.3 Previous Works 8

1.3.1 Physical Snake Robot 8

1.3.2 Control of the Snake Robot 10

1.4 Overview 14

1.4.1 Scope of the Thesis 14

1.4.2 Outline of the Thesis 16

2 Simulation and Experiment Platform **18**

2.1 ODE-based Simulation Platform 19

2.2 Experiment Platform - *SR*-I 20

2.2.1 Physical Prototype 20

2.2.2 Control System . . . 23
2.3 Summary . . . 24

3 CPG Model and Network **26**
3.1 Central Pattern Generator . . . 27
3.1.1 Neural Model . . . 28
3.1.2 Comparison of Two CPG Models . . . 29
3.2 CPG Network with Feedback Connection . . . 30
3.2.1 Mathematical Model of CPG Network . . . 32
3.2.2 Unilaterally-connected Network . . . 32
3.2.3 Closed-loop Network with Feedback Connection . . . 33
3.2.4 Comparison of Two CPG Networks . . . 34
3.3 Characteristic of CPG network . . . 36
3.3.1 CPG Parameters . . . 36
3.3.2 Number of the Locomotion S-shape . . . 37
3.4 Summary . . . 39

4 CPG-based Motion Control **40**
4.1 Basic Motion Patterns . . . 40
4.1.1 Curvature of Locomotion . . . 40
4.1.2 Speed of Locomotion . . . 42
4.1.3 S-shape of Locomotion . . . 42
4.2 Motion Optimization . . . 43
4.2.1 Curvature Adaptive Principle . . . 43
4.2.2 Ground with Different Frictions . . . 46
4.2.3 Slope with Different Inclined Angles . . . 48
4.2.4 Discussion of Ratio μ_n/μ_t . . . 50
4.3 Summary . . . 53

5 CPG-based Self-adaptive Control **54**
5.1 Adaptive Motion Control . . . 54
5.1.1 Turn motion . . . 55
5.1.2 Round motion . . . 58
5.1.3 Head-navigated locomotion . . . 61

5.1.4 Experiments . . . 66
5.2 Modeling of AMM-based Turn Motion . . . 69
5.2.1 Amplitude Modulation Method . . . 69
5.2.2 Computational Model . . . 70
5.2.3 Features of AMM . . . 72
5.3 Sensor-driven Neural Controller . . . 75
5.3.1 Design of Collision-free Behaviors . . . 75
5.3.2 Design of Neural Controller . . . 78
5.3.3 Neural Preprocessing Network . . . 79
5.3.4 Preprocessing Neuron for Sensory Signals . . . 82
5.4 Experimental Analysis . . . 83
5.4.1 Autonomous Reactive Behavior . . . 83
5.4.2 Simulation Result . . . 84
5.4.3 Experimental Result . . . 85
5.5 Summary . . . 86

6 Conclusion and Future Work **89**
6.1 Conclusion . . . 89
6.2 Future work . . . 90

Reference **92**

A Control System **98**

B Circuit Design **100**

List of Figures

1.1 Snakes in nature can perform various motion patterns to adapt with the environment. 1

1.2 Anatomy of snakes (Image from [51]) 5

1.3 Scales covering on the snake body (Image from Banded Krait in [49]) 6

1.4 Various motion gaits of the snake (Image from Hirose in [19]) 7

1.5 Wheel-based snake robots. (a)Hiroses Active Cord Mechanism utilized a series of articulated links with passive wheels; (b)The AmphiBot II robot designed for both serpentine locomotion and swimming; (c)A lifelike snake robot prototype S5 by Dr. Gavin Miller; (d)Amphibious snake-like robot Perambulator by Prof.Ma; (e)A caterpillar drive robot with a snake-style body can climb over small and large obstacles 9

1.6 Wheel-less snake robots. (a)Hiroses Slim Slime Robot utilized a series of pneumatically-driven modules; (b) An obstacle-aided snake robot: Aiko; (c)A snake robot demonstrates how it can climb a tree; (d)Polychaete-like undulatory robotic locomotion in unstructured substrates 10

1.7 CPGs are neurons existing in the spinal cords of living animals which are used as activation signal for muscle contractions. 12

1.8 CPGs generate purposive rhythmic patterns by using the interaction between animals and sensory environment. 13

2.1 Simulation platform of the snake-like robot 20

2.2 Overview of our snake-like robot 21

2.3 Structure of one link in *SR*-I 22

2.4 Arrangement of the sensors in *SR*-I 22

2.5 Structure of control system 23

2.6 Flow chart of communication 24

3.1 CPG models of neural oscillator 28

3.2 Outputs of two CPG models and their derivatives 30

3.3 CPG network implemented to control a snake-like robot 31

3.4 Schematics of different CPG connection networks 31

3.5 Rhythmic outputs of two kinds of CPG models using unilaterally-connected network 33

3.6 Rhythmic outputs of two kinds of CPG models in closed-loop network . . . 35

3.7 Results to show the relation between the CPG parameters and the CPG output 37

3.8 Oscillation signals changed with respect to Driving input and Time constant where (a) Driving input $u_0 = t/5 + 2.5$; (b) Time constant τ_1=2.0, 3.5, 5.0 and τ_2=6.0, 7.5, 15.0 for time t=0, 60, 120 (s) 38

4.1 Locomotion of the robot with different body curvatures 41

4.2 Curvature with respect to CPG driving input 41

4.3 Motion speed with respect to CPG time constant 42

4.4 Numbers of S-shape changed through feeding a different CPG back to the first one 43

4.5 Curvature adaptive principle 45

4.6 Simulation results under different friction 47

4.7 The relation between optimal u_0 and friction coefficient μ_t 48

4.8 Simulation results under different incline angle 49

4.9 The relation between optimal u_0 and incline angle ϕ 50

4.10 Scenes of the creeping motion on the slope with different driving input . . . 51

4.11 Trajectories of robot under four different friction 52

4.12 The relation between optimal u_0 and friction coefficient 52

5.1 Schematic of the parameter transmission network. 55

5.2 Turning motion with asymmetric locomotion (a) The angle signals of joints (b) The trajectory of the snake robot 56

5.3 Simulation for turn motion 57

5.4 Turning angle with respect to different bias Δu_0 58

5.5 Round motion with asymmetric locomotion (a) The angle signals of joints (b) The trajectory of the snake robot 59

5.6 Simulation for round motion 60

5.7 Maximum distance of the motion curve with respect to different bias $\Delta\tau$ . . 60

5.8 Head-navigated locomotion of the snake-like robot 61

5.9 Configurations of the joint angles in the head-navigated locomotion of the snake-like robot 63

5.10 CPG network implemented to control a head-navigated locomotion 65
5.11 Simulation results of the head-navigated locomotion 65
5.12 Control signals of the joints in the simulation. Red line and blue line represent the angle signal of the first joint in two different motion patterns. Black lines are the angle signals of other joints. 66
5.13 Experiment scenes of the snake-like robot to avoid obstacle with turn motion 67
5.14 Experiment scenes of the snake-like robot to avoid obstacle with round motion 67
5.15 (a) Experiment scenes of the head-navigated locomotion; (b) Experiment scenes of the normal serpentine locomotion 68
5.16 Turn motion of the snake-like robot. (a) The angle signals of joints (b) The trajectory of the head link during the locomotion 70
5.17 Gliding curve of the snake creeping locomotion 71
5.18 Features of the AMM-based turn motion 73
5.19 An example of right turning motion in two different cases 74
5.20 Circular motion trajectory of the turn motion 75
5.21 Sensor arrangement with head-navigated locomotion 76
5.22 Scheme for the motion strategies in different situation. LS: Left Sensor, FS: Front Sensor, RS: Right Sensor, LT: Left Target, RT: Right Target 77
5.23 Flow diagram of the sensor-driven neural controller 79
5.24 Structure of the Neural Preprocessing Network. LT: Left target command; RT: Right target command; LS: Left sensory signal; FS: Front sensory signal; RS: Right sensory signal; LMN and RMN: Left and right middle neuron; LON and RON: Left and right output neuron; T: Threshold values of neurons; PD: Phase delay; g: Gain coefficient 80
5.25 An example of the results of the neural output from the NPN 81
5.26 Preprocessing Neuron for Sensory Signals. (a) Structure of the scaling model; (b) Output of waves after preprocessing; a=7.2;b=6;c=6 83
5.27 (a) Schematic of simulation environment for the collision-free behavior. (b) Screenshot of the simulation 84
5.28 Trajectory of the simulation results. (a) Moving in clockwise. (b) Moving in counterclockwise 85
5.29 Experiment scenes of the collision-free behavior 88

A.1 Architecture of control system for the snake-like robot 99
A.2 Schematic block diagram of the main program in the environment of Simulink/Matlab 99

B.1 The developed controller with dsPIC-based Microprogrammed Control Unit 100

B.2 Electronic schematic of the designed circuit 101
B.3 Finished circuit board for each snake joint 102

List of Tables

2.1 Physical parameters of the simulated snake robot 20
2.2 Mechanical specifications of the robot 21

3.1 Parameters of the CPG model 30
3.2 Change of the CPG output with respect to each CPG parameter 38

5.1 The relation between the direction of robot and the controlled bias 56
5.2 State Table of the neurons with respect to different input 82
5.3 Physical parameters of the simulated robot 84

Acknowledgements

At the time I am finishing the doctoral course in Ritsumeikan, I would like to acknowledge many people who have provided me with great help and much attention.

I would like to express my deep and sincere gratitude to my supervisor, Professor Shugen Ma, a professor in Department of Robotics at Ritsumeikan University, whose wide knowledge and his logical way of thinking have been of great value for me. His encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the robotics. He not only gives me great help on research, but also takes care of me on daily life. Although I will complete the doctoral course soon, a heart with thanksgiving will exist in my life forever.

I would like to express my appreciation to Professor Sadao Kawamura and Professor Masaaki Makikawa who are professors in Department of Robotics at Ritsumeikan University. During the study in university, they have given me valuable suggestions on the research and presentation. Thanks for their comments on this thesis for a doctorate.

It is my pleasure to thank my colleagues in Ma laboratory. I feel very lucky to work with all of them. Thank my friends in Ritsumeikan University. Studying with them together in Ritsumeikan, I have shared a colorful campus life along with them.

I am grateful to the professors in my mother university, China University Of Petroleum. Seven years' study for my bachelor and master degree will be precious memories.

Thank the China Scholarship Council. During the three years, they supported me on finance so that I can concentrate on my doctoral research without doing any part-time job.

Without the support and understanding of my family, it is impossible for me to complete this doctoral dissertation. I love them.

Curriculum Vitæ

Xiaodong Wu

Education

1990-1995	Zhangmuqiao Primary School (Zhangmuqiao, Shucheng, Anhui, China) Pupil
1995-1998	Zhangmuqiao Junior High School (Zhangmuqiao, Shucheng, Anhui, China) Junior School Student
1998-2001	Shucheng Middle School (Shucheng, Anhui, China) High School Student
2001-2005	China University Of Petroleum (Beijing, China) Bachelor, Machine Design, Manufacturing and Automation
2005-2008	China University Of Petroleum (Beijing, China) Master of Engineering, Mechatronic Engineering
2008-2011	Ritsumeikan University, BKC (Kusatsu, Shiga, Japan) Doctor of Philosophy, Robotics

Personal

Born	October 26, 1984, Anhui, China.
Interests	Photography, Badminton, Music, Travel, Programming.

Chapter 1

Introduction

1.1 Research Background

Snake living widely across the earth is a fascinating part of the natural world. Its particular limbless motions and physiology make it has ability to move in various environments which are shown in Fig. 1.1. These amazing creatures are optimal in the sense that they have emerged through millions of years of evolution. It is hoped that by imitating from biological snake, man-made robots can be developed with unusual capabilities as well as answering questions about how snakes in nature really move.

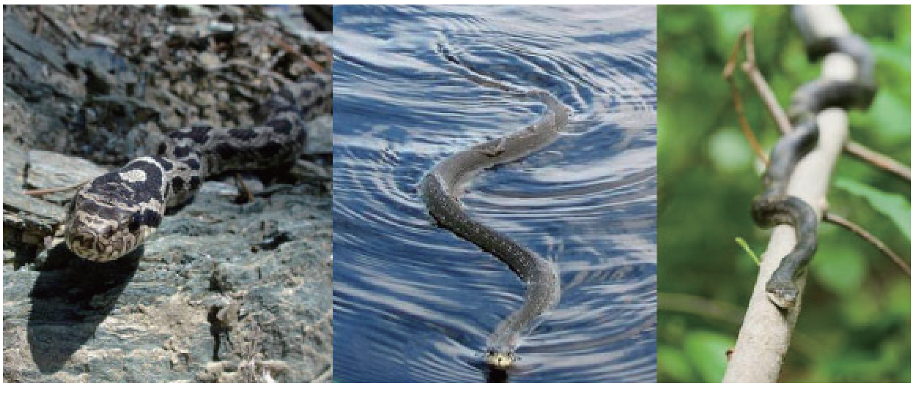


Figure 1.1. Snakes in nature can perform various motion patterns to adapt with the environment.

Design of snake robots with the characteristics of natural snakes is challenging both in mechanical design and control design. One of the fundamental issues is understanding the motion patterns of snake. In general, propulsion in wheeled and legged robots is more readily understandable versus the locomotion of limbless snake, which is move forward by the swing of the body joints from side to side. It is believe that the skeletal structure and the special scales are main contributions to the limbless snake locomotion.

By utilizing the useful features of serpentine locomotion, a man-made snake robot will be expect to have the ability to move into unstructured environment where is difficult for the access of traditional wheeled or legged robots. A snake-like mechanism that could slide, wiggle and even swimming is possible to be developed for many kinds of applications, like exploration, medical device, and so on.

1.1.1 Applications

Why do we want to develop snake robots? Consultation with potential users, and examination of many application areas suggests a number of areas where serpentine robots can make an impact [9]. For example, robotics are expected to be applied in areas that include earthquake search and rescue, nuclear plants, medical applications and the inspection of hard-to-reach areas. Each of the following description offers a potential application for the utilization of the serpentine robot. These applications mostly have some difficulties and problems by using of wheeled or legged robots, where snake-like robot or mechanism could perform well.

Exploration

Human activity is precluded in many unaccessible areas include planetary surfaces, earthquake field and extreme terrains with loose rubble. A snake-like device can distribute its mass over a large area for support so that it can still move effectively. Self-support between larger numbers of contact points enables continued operation and movement.

Inspection

Many inspection devices in industry rely on fixed-base mechanisms such as borescopes, videoscopes and fiberscopes. These devices are primarily used to inspect unaccessible area that cannot be seen directly by the eye. Simple direct-view borescopes have proven useful, but articulated self-advancing devices which can follow complex paths could open many more applications. Due to the limitations of traditional methods, developers of inspection equipment are keenly interested in self-propelled inspection devices, like snake-like robots. With a serpentine tool, inspections and accurate localization could have lower costs and downtime significantly.

Medical

Recently non-invasive surgery has met with wide acceptance and produced remarkable results. Laparoscope devices, which are rigid tools inserted into the abdominal wall, and endoscopic devices are used in these types of surgical procedures. These surgical tools, however, are often difficult to operate and have their limitations. Snake-like devices have also received attention as a potential medical mechanism. A snake-like robot could be a self-propelled endoscopic device for more challenging surgery.

Reconnaissance

Concealment and reconnaissance offer some novel applications of serpentine mechanisms. The ability to command small roving eyes and ears makes it be a potential application by law enforcement agencies. The snake-like robots can camouflage themselves very easily and move through the underbrush where is difficult for other robots. This ability could also be used for military reconnaissance. A snake-like robot has already been studied and utilized in the Israel Defense Forces.

1.1.2 Challenges of Limbless Locomotion

While the advantages and the applications for serpentine robots are attractive, there remain many challenges in realizing such limbless locomotion. To create a truly successful snake robot requires that many areas be addressed and solved.

The challenge of configuration and design of the whole robot is determining the technology that drives the mechanism. The dependence on environment interaction is more complicated for a snake robot than for conventional mobile robot. A snake robot, on the other hand, has no separate part which is dedicated to propulsion. Being essentially a smooth and flexible manipulator arm, the propulsion mechanism of a snake robot is rather an integrated part of the entire body, which means that propulsion requires synchronized motion of the entire robot in order to produce appropriate environment interaction forces. Motion based on such environment interaction is challenging both with respect to control design and mechanical design.

Furthermore, a snake robot has many degrees of freedom, which means that the physical mechanism will contain a complex interconnection of sensors, actuators, and control logic. The many degrees of freedom represent complex nonlinear dynamic which is challenging to analyze from a control design perspective.

Moreover, selection of the actuators and design of control system in robots with normal size are not considered as a hard task. However, Serpentine robots must be compact and small to guarantee for the advantages shown in the previous section. Small size burdens the tasks of wire routing and actuation support.

1.2 Biological Snakes

The difficulty in the realization of snake robots is how to study from biological animals to man-made mechanisms. It is important to investigate the anatomy of snakes in nature, and use the animal as a model or blueprint for robotic mechanisms. In the following, we present aspects of biological snakes that we consider relevant to development of snake robots.

1.2.1 Anatomy of Snakes

The typical appearance of the skeletal structure of a snake is shown in Fig. 1.2. Snake in nature can have between 130 and 500 vertebrae, with ribs attached to each other. Snake vertebral articulation is one of the most complex of all vertebrates. Although only a few limited motions and amplitudes are possible between adjacent vertebrae, concatenation of these articulations can produce large angular signals. The vertebrae of snakes are quite simplified in number and structure in comparison to other limbed vertebrates. This kind of repeated structure with relatively limited motions between adjacent vertebraes are worth to investigate in a mechanism design.

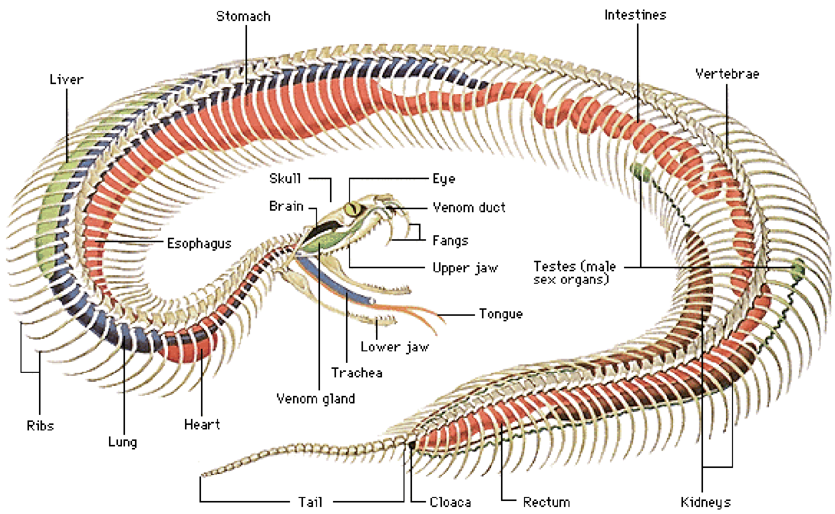


Figure 1.2. Anatomy of snakes (Image from [51])

The skin of a snake is completely covered with scales shown as Fig. 1.3. An important purpose of the scales is to form a physical protection from general wear and tear when the snake moves across rough surfaces. Another feature of the scales is that they give the snake anisotropic friction properties with the ground. The scales can have a larger friction coefficient in the transversal direction of the snake body compared to in the tangential direction. The assumption on this important friction property is widely used in the studied of the biological snake robots.

In this study, it is not possible to replicate all of above capabilities in sensing and

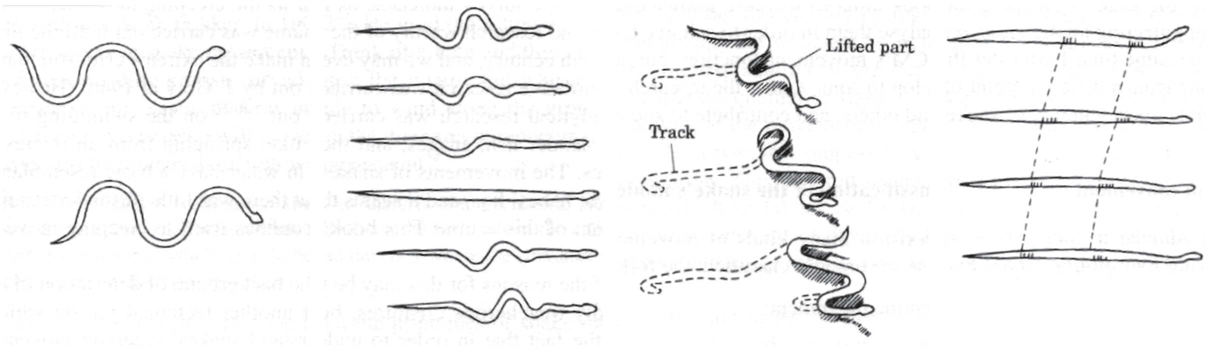
Figure 1.3. Scales covering on the snake body (Image from Banded Krait in [49])

control. There are few commercial mechanisms like physiological properties of snakes, such as larger number of vertebraes, special skin and scales. By using simplified model and bio-inspired methodology, this study shows that it is possible to replicate general characteristics of serpentine locomotion.

1.2.2 Locomotion of Snakes

Despite of the lack of legs, it do not appear to have placed restrictions on the ability of snakes to move around. On the contrary, snake can move with various motion patterns flexibility. There are significant studies on the qualitative aspects of snake locomotion. Several categories of snake-like locomotion are shown in Fig. 1.4, including serpentine crawling, concertina, sidewinding, rectilinear, slide-pushing and other less common forms [15]. These gaits are performed depending on different environment to obtain an optimal movement.

Serpentine crawling, also called lateral undulation is the fastest and most common form of snake locomotion. Continuous waves are generated along the snake body from the head to tail during the lateral undulation. The same wave form is produced in swimming gait. During this motion pattern, the sides of the snake body push against the contact ground due to the asymmetric friction, thereby pushing the snake forward. Lateral undulation requires



(a) Serpentine (b) Concertina (c) Sidewinding (d) Rectilinear

Figure 1.4. Various motion gaits of the snake (Image from Hirose in [19])

a minimum of three contact points for continuous forward progress: two to generate force and the third to balance forces to move in a particular direction [16]. Lateral undulation has a high motion efficiency on the flat ground, but not suitable on slippery surfaces.

Concertina gait is usually employed in narrow areas, such as tunnels and pipes where the snake cannot utilize the full amplitude of body to perform lateral undulation. This gait provides a base in which parts of the body stop for purchase and other parts move forward. The sequence repeats, and the snake moves forward. The principle behind concertina locomotion is the utilization of the difference between the large static friction forces and low kinetic friction forces in different part of the body [68]. This motion gait is not efficient compared to serpentine crawling.

Sidewinding is used primarily by snakes that lives in desert regions where the ground is covered with loose soils and sands. Starting from a resting position, the head and neck are raised off the ground and thrown sideways while the rest of the body provides an anchor against the ground. Once the front part of the body are on the ground, they in turn act as an anchor while rest of the body repeats the same motion. This motion pattern can minimizes slippage and is proven to be even more efficient than lateral undulation [57].

Rectilinear movement is used by many snakes to climb trees or move through narrow burrows. In addition, many heavy-bodied snakes may use rectilinear movement when they are crawling on the ground. During rectilinear motion, the snake uses the edges of the scales on its underside as anchor points to pull itself forward in a more or less straight line.

More specialized forms of limbless locomotion also exist. For instance, certain snakes can jump by making their body into a vertical S-shape, and then jump by stretching the body rapidly. Some snakes can leap gaps of a meter or more [19]. An optimal shape of the grounded part of a snake robot with passive wheels for jumping has been discussed in [26]. Other extraordinary modes are used by certain asian tree snakes [61] that can fly through the air by throwing themselves from trees and forming their body in an aerodynamical shape.

1.3 Previous Works

One of the earliest analytical studies of snake locomotion was given by Gray in [15]. Mathematical descriptions of the forces acting on a snake are proposed and used to derive properties of snake locomotion. After that, many kinds of the eel-like or snake-like robots presenting elegant mechanical designs or life-like movement have been created. These snake-like robots were controlled to imitate the observed serpentine locomotion. Previous literature that considers development of snake robots is summarized in the following.

1.3.1 Physical Snake Robot

Hirose developed the world's first snake robot as early as 1972 [19]. This robot was equipped with passive wheels to realize the anisotropic ground friction property that enables forward locomotion on flat surfaces. Several other snake robots with passive wheels have been proposed over the years, such as the robots presented in [11] [65] [36] [71] [35] [45] [84] [81] [7] [86] [27]. Some of the robots can only display planar motion, while other robots can move their links both horizontally and vertically. Some robots have shielded joint modules that enable motion in environments with e.g. mud and dust, and even motion under water, while other robots have modules with exposed electronic components which only allow them to move in clean lab environments. A common feature of these mechanisms is that they are generally only able to move on relatively flat surfaces since passive wheels are not suitable to move very well in a cluttered environment. Active wheels has been used in a tracked snake robot to solve this problem [3]. However, the motion efficiency of the lateral undulation by passive wheel-based snake robots on the flat ground is much higher than other snake robots.

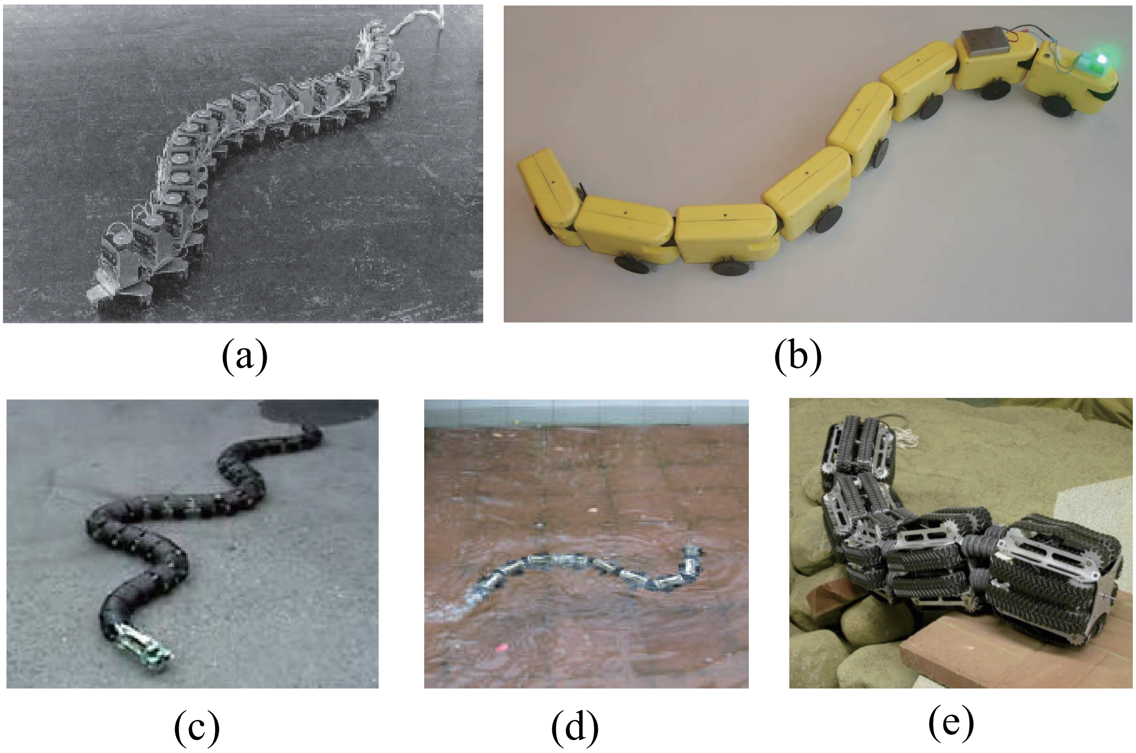


Figure 1.5. Wheel-based snake robots. (a)Hiroses Active Cord Mechanism utilized a series of articulated links with passive wheels; (b)The AmphiBot II robot designed for both serpentine locomotion and swimming; (c)A lifelike snake robot prototype S5 by Dr. Gavin Miller; (d)Amphibious snake-like robot Perambulator by Prof.Ma; (e)A caterpillar drive robot with a snake-style body can climb over small and large obstacles

Snake robots without wheels, which basically consist of straight links interconnected by motorized joints, are presented in [85] [72] [9] [46] [48] [55] [4] [32] [82] [47] [62]. Despite its lack of wheels, the snake robot in [55] maintains an anisotropic ground friction property since the underside of each link has edges, or grooves, that run parallel to the link. Other robots whose ground friction properties are isotropic, on the other hand, can move forward during lateral undulation by slightly lifting the peaks of the body wave curve from the ground which is called sinus-lifting gait. Some of them can traverse cluttered and irregular environments by using irregularities around its body as pushpoints to aid the propulsion [2] [66]. This is denoted obstacle-aided locomotion and requires snake robot to have a smooth exterior surface combined with a contact force sensing system. However, robots with isotropic friction are mostly used for studying gaits other than lateral undulation, such as gaits based on sidewinding, inchworm motion, or lateral rolling.

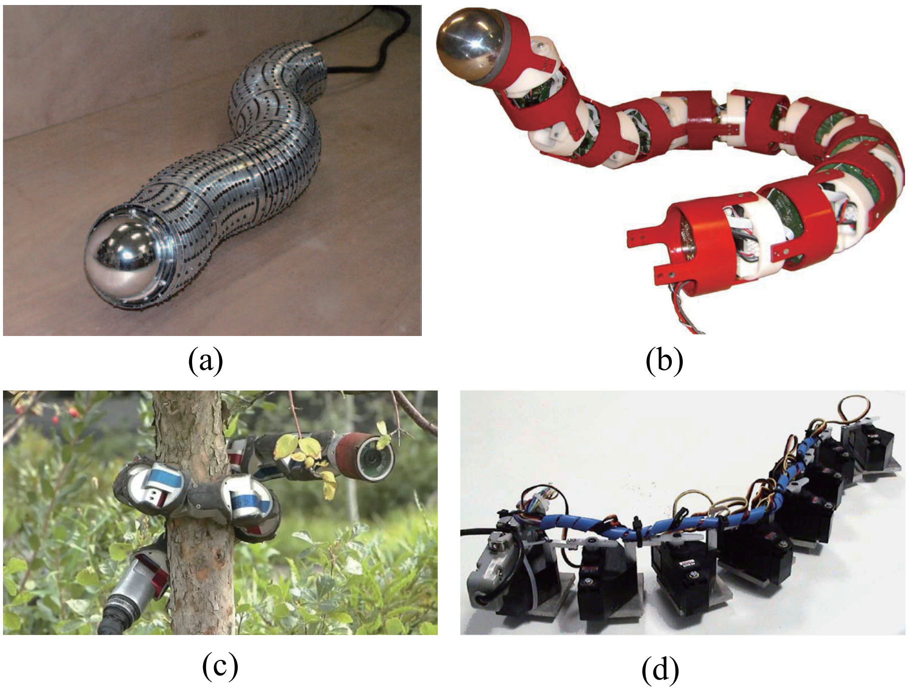


Figure 1.6. Wheel-less snake robots. (a)Hiroses Slim Slime Robot utilized a series of pneumatically-driven modules; (b) An obstacle-aided snake robot: Aiko; (c)A snake robot demonstrates how it can climb a tree; (d)Polychaete-like undulatory robotic locomotion in unstructured substrates

1.3.2 Control of the Snake Robot

Development of the control system of a snake robot is generally quite challenging work. As we know, a snake robot has many degrees of freedom, which means that a complex interconnection of sensors, actuators, and control logic needs to be handled. The high degree of freedom with complex dynamics is challenging to analyze from a control design perspective. A considerable number of snake-like robots have been constructed in the past.

These snake-like robots were controlled to imitate the observed motion patterns of natural snakes with elegant control designs. Their control architecture can roughly be divided into three categories: sine based, model based, and CPG based.

Sine-based approaches

Sine-based approaches use simple time-indexed sine-based functions for generating traveling waves (for example, [45] and [67]). The advantages of such an approach are its simplicity and the fact that important quantities such as frequency, amplitude, and wavelength are explicitly defined. A disadvantage is that online modifications of the parameters of the sine function (e.g., the amplitude or the frequency) will lead to discontinuous jumps of setpoints, which will generate jerky movements, risking damages of the motors and gearboxes. This problem can, to some extent, be overcome by filtering the parameter and/or the outputs, but the approach then loses its simplicity. Another disadvantage is that sine-based functions do not offer simple ways of integrating sensory feedback signals.

Model-based approaches

Model-based approaches use kinematic [52] [38], or dynamic [43] models of the robot to design control laws for gait generation. The model-based approaches offer a way to identify fastest gaits for a given robot by using kinematic constraints or approximations of the equations of motion. Therefore this kind of approach is very useful for helping to design the controllers. But it still has two limitations. First, the performance of controllers will deteriorate when models become inaccurate, which is rapidly the case for interaction forces with a complex environment (e.g., friction with uneven ground). Second, if the mechanism of the robot is changed a little, the control model needs to be redesigned. The resulting controllers are not always suited for a universal application.

CPG-based approaches

CPG-based approaches, which is our method, is inspired by a control mechanism found in animal bodies and the requirement to easily modulate locomotion: Central Pattern Generators (CPGs). CPGs are neural networks existing in the spinal cords of living animals which are used as activation signal for muscle contractions in animals. It is capable of producing coordinated patterns of rhythmic activity without any rhythmic inputs from sensory feedback or from higher control centers [8]. Even completely isolated CPGs in a petri dish can produce patterns of activity, called active locomotion, that are very similar to locomotion activated by simple electrical or chemical stimulation [17]. CPGs generate a purposive rhythmic motion pattern flexibly by using the entrainment caused by inter-CPG interaction and the sensory input from the outside environment. From a control point of view, CPGs can be implemented on the control of some kinds of biological robots for rhythmic motions, like walking, swinging, and flying.

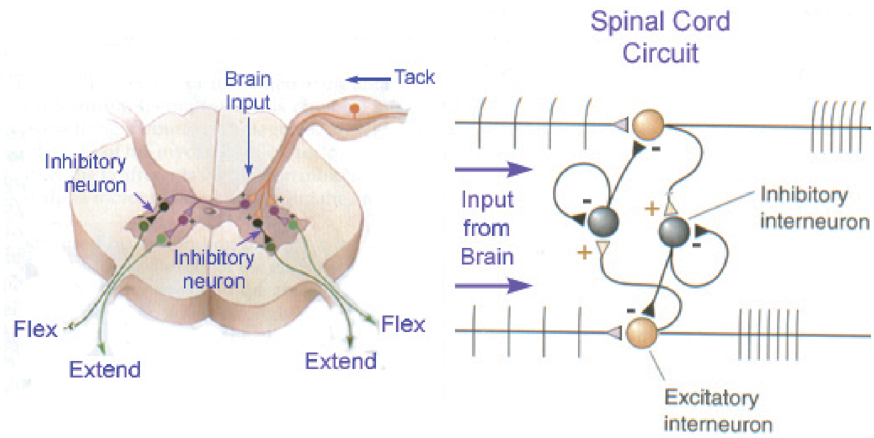


Figure 1.7. CPGs are neurons existing in the spinal cords of living animals which are used as activation signal for muscle contractions.

CPG-based approaches use dynamical systems coupled nonlinear oscillators or neural networks to generate the rhythmic waves necessary for locomotion. Generally, the advantages of the CPG-based controller can be summarized into three parts. 1) Robustness: These approaches are implemented as differential equations integrated over time, and the goal is to produce the travelling wave as a limit cycle. If this is the case, the oscillatory

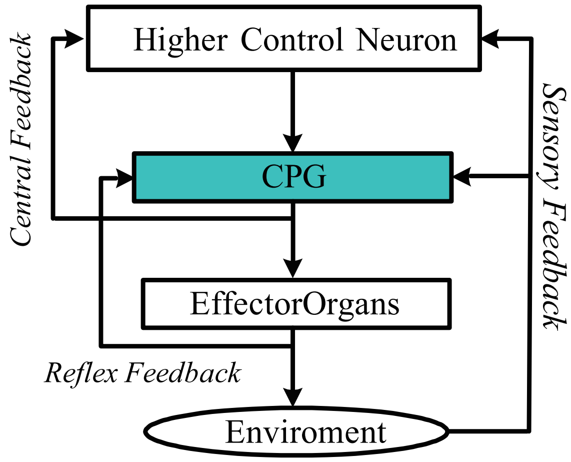


Figure 1.8. CPGs generate purposive rhythmic patterns by using the interaction between animals and sensory environment.

patterns are robust against transient perturbations due to the characteristic of the limit cycle. 2) Real-time: Furthermore, the nonlinear model can usually be modulated by some parameters, which offer the possibility to smoothly modulate the type of gaits produced. It could be convenient to get desired rhythmic motions by changing the parameters of CPGs in real time. 3) Adaptability: One important feature of the CPG system is that it can not only receive inputs from a higher level of the central nerve system, but also from peripheral receptors [42]. CPGs can readily integrate sensory feedback signals in the differential equations and show interesting properties such as entrainment by the mechanical body [70] [63].

Due to the advantage of the CPG in rhythmic motion control, many researchers have placed CPG schemes into the control of robots. By implementing CPG oscillators in the mechanism of a robot arm, more adaptive movement was performed [70]. A dynamic gait was achieved through using a neural system model on a quadruped robot [13]. Based on biomimetic central pattern generators and on information from distributed distance sensors, neuromuscular motion control for an undulatory robot models was presented [59]. An amphibious snake robot realized a crawl motion by utilizing an on-board central pattern generator [7]. The generation of rhythmic and voluntary patterns of mastication has been tested on a humanoid chewing robot by using the CPG oscillator [80].

This kind of CPG-based mechanism is quite suitable for the control of a snake-like robot with large DOFs, where each joint of the segment has at least two DOFs. However, one difficulty with CPG-based approaches is to determine how to design the CPG to produce a particular pattern. Many CPG models do not have explicit parameters defining quantities such as frequency, amplitude, and wavelength (for instance, a van der Pol oscillator does not have explicit frequency and amplitude parameters). In this thesis, we use a CPG model based on Matsuoka's amplitude-controlled phase oscillators. An interesting aspect of this approach is that the limit cycle of the CPG has a closed form solution, with explicit frequency, amplitude, and wavelength parameters. The approach, therefore, combines the elegance and robustness of the CPG approaches with the simplicity of sine-based approaches. Furthermore, our CPG model is computationally very light, which makes it well suited to be programmed on a microcontroller on board of the robot. By coupling the sensory signals with the CPG neural oscillators, an adaptive serpentine locomotion of the snake-like robot was achieved in the study.

1.4 Overview

1.4.1 Scope of the Thesis

The work underlying this thesis has been carried out with the following scope.

Sensor-based snake-like robot

A biologically inspired snake-like robot has been widely studied for its unique motion patterns. To achieve self-adaptive locomotion, a sensor-driven snake-like robot *SR*-I has been developed. Such robot can achieve self-adaptive locomotion by coupling sensory information with the motion control system. The design of the mechanism and control system of this sensor-driven snake-like robot is presented in the paper. Based on the sensory information, the proposed controller is implemented on the *SR*-I to verify the efficiency of the algorithm.

CPG-based neural controller

A CPG-based approach is proposed to solve the difficulty in control of a snake-like robot with a large number of degrees of freedom. Compared with the previous research, a new network with feedback connection is presented, which can generate uniform outputs without any adjustment. Furthermore, the relation characteristics between the CPG parameters and the outputs are investigated. Based on the analyzed results, desired rhythmic outputs (for instance, different amplitude or period) can be achieved by adjusting the CPG parameters with respect to human command.

Motion control

Based on the analysis of the CPG neural network, the motion patterns of the snake-like robot have been realized by modulating corresponding parameters. The control of body amplitude, speed, and number of S-shape during the locomotion is investigated respectively by utilizing the properties of CPG parameters. Both simulation and experimental results of the snake-like robot have been taken for the analysis of the locomotion control. Furthermore, to get motion optimization of the snake robot, a curvature adaptive principle is proposed. This principle has been used to evaluate adaptive creeping locomotion of the robot to changed friction and given gradient.

Self-adaptive control

Most of the studies of the control of the snake-like robot employed the methods based on derivation of purposive body motion to design specific controllers. However, there appears to have been relatively limited work on sensor-based closed-loop motion control of the snake robot. To achieve autonomous reactive behaviors, a sensor-based neural controller is presented in this paper. Such controller can achieve self-adaptive locomotion by tightly coupling sensory information to the corresponding body action, without necessarily the intervention of elaborate world model. Due to the nonlinear property of neural module, the motion transition of the snake robot can be achieved smoothly in real time, by the means of the modulation of the CPG oscillation parameters. By fusing the peripheral signals through a proposed neural preprocessing network, a driving input will be generated onto the CPG oscillator network to conduct a desired reactive behavior. The performance of the proposed sensor-driven neural controller was verified by conducting an experiment on a snake robot in an environment with obstacles.

1.4.2 Outline of the Thesis

The purpose of this research is to develop a neural oscillator controller for the serpentine locomotion of a snake-like robot. The contents of this thesis are organized as follows.

Chapter 2 introduces a simulation platform and a sensor-driven snake-like robot *SR*-I. The hardware configuration and software design are both presented. The proposed control system in other chapters will be verified by the simulation and physical experiment of *SR*-I.

Chapter 3 proposes a bio-inspired system imitating the neural oscillators to drive the serpentine locomotion of the snake robot. The model of the neural oscillator network is designed with nonlinear equations. The influences of model parameters on the rhythmic output are investigated and the relation curves are obtained. The design of motion control is discussed by the results of these relationships.

Chapter 4 presents the principle of the CPG-based motion control. Three typical motion parameters: motion curvature, speed, and numbers of *S*-shape are controlled by the proposed neural oscillator network. Furthermore, to evaluate optimal creeping locomotion of the robot, an energy-based motion optimization of the snake-like robot is also taken into discussion.

Chapter 5 is mainly focused on the adaptive control of the snake-like robot. Firstly, turn motion, round motion and head-navigated motion are introduced respectively as the preparation work of following adaptive control. After that, a detail modeling of the AMM-based turn motion is concluded. Based on these results, a sensor-based neural controller is proposed to self-adaptive reactive behavior of a snake-like robot. The designed neural controller can fuse multi-signal to generate driving input for CPG network and activate an appropriate reactive behavior. The proposed model is verified by a collision-free behavior of the snake-like robot *SR*-I.

Chapter 6 concludes this thesis and discusses the limitation of the existing works. Some possible works in the future are described.

Chapter 2

Simulation and Experiment Platform

In recent years, many studies have been carried out to realize an environment-adaptive locomotion of the snake-like robot. Some of them are based on ingenious mechanism design to adapt to the surroundings passively, including wheel-less snake-like robots for an obstacle-aided locomotion in [66][2], and a snake-like robot utilizing passive joints in [25]. Others, which are also dominant, rely on an operator to control the robot actively to adapt to the unstructured environment. Energy-based control method for the snake-like robot to adapt to different friction and slope was also proposed in [75]. Crespi and Ijspeert take online optimization of swimming and crawling in an amphibious snake robot by neural control method [7]. However, the above methods are all based on the motion planning or human controlled system without the integration of sensory information [78].

To enhance the autonomy of the snake-like robot, researchers have also proposed many kinds of sensor-based solutions. One of the first such explorations appears to be the ACM-III wheeled snake robot, which is equipped with tactile sensors along its body and demonstrates movement inside narrow corridors and adaptive coiling around objects [19] [20]. Paap and Christaller [53] realized a semi-automatic motion of the snake robot by scanning the environment with a video camera and two ultrasonic sensors. Sfakiotakis and Tsakiris [58] used numbers of distance sensors to collect the information of surroundings and navigate an undulatory robot in a complex corridor.

The goal of this research is to propose a neural oscillator controller for a snake-like robot. To verify the control system, simulation and physical experiment platform is necessary. Thus a 3-dimensional dynamic simulation environment has been developed. For physical experiment, a sensor-driven snake-like robot *SR*-I is designed for the verification of proposed controller.

2.1 ODE-based Simulation Platform

To verify the proposed control method, a simulator for a snake-like robot has been developed in an Open Dynamics Engine (ODE) environment. ODE is an open source, high performance library for simulating rigid body dynamics. It is fully featured, stable, mature and platform independent with an easy to use C/C++ API. It has advanced joint types and integrated collision detection with friction. ODE is useful for simulating robots, objects in virtual reality environments and virtual creatures [60].

The designed simulation of the snake-like robot is shown in Fig. 2.1. In the simulation, the interaction between the robot and the ground is modeled with asymmetric friction by using a larger normal friction coefficient μ_N and a smaller tangential friction coefficient μ_T. To realize this kind of friction model, a passive wheel is utilized for each link of the snake-like robot. The actuators are installed on the joints of the robot to make each joint swing from side to side, like the behavior of a natural snake. The physical parameters of our snake-like robot platform are given in table 2.1.

During the simulation, each joint of the robot is controlled by an oscillator. The rhythmic outputs of the oscillators are used as angle inputs of the joints. The angle signal of the robot joint θ_i can be calculated by

$$\theta_i = \alpha_i y_{out,i} \tag{2.1}$$

where α_i is a gain from the control signal to the joint angle; $y_{out,i}$ is the output from i-

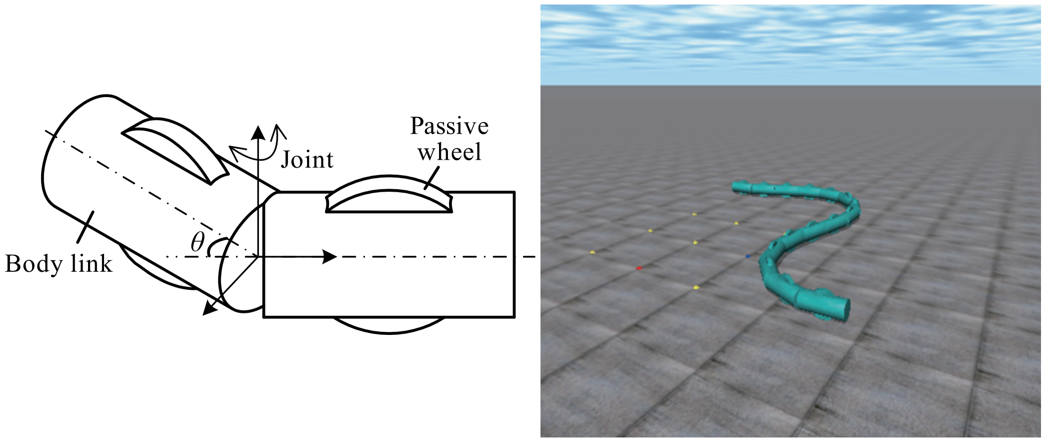


Figure 2.1. Simulation platform of the snake-like robot

Table 2.1. Physical parameters of the simulated snake robot

Numbers of joints :	Num=10
Length of link:	L_{link}=0.13m
Radius of link:	R_{link}=0.02m
Weight of link:	M_{link}=0.2kg
Width of wheel:	L_{wheel}=0.01m
Radius of wheel:	R_{wheel}=0.03m
Weight of wheel:	M_{wheel}=0.08kg
Friction coefficients:	μ_N=0.5 μ_T=0.02

th oscillator. By a series of successive rhythmic signals added on the robot, a snake-like serpentine locomotion can be performed in the designed simulation environment.

2.2 Experiment Platform - *SR*-I

2.2.1 Physical Prototype

The overview of our snake robot *SR*-I is shown in Fig. 2.2. The robot model is composed of 10 segments which are connected serially. Each segment is connected by joints with 2 DOFs which can rotate on a vertical axis (yaw) and a horizontal axis (pitch). The whole

Figure 2.2. Overview of our snake-like robot

length of the robot is about 1.3 m. The mechanical specifications of the robot are shown in table 2.2.

Table 2.2. Mechanical specifications of the robot

Items	Details
Numbers of joints :	20
Size of link[mm^3]:	130×62×77
Weight of link[kg]:	0.28
Motion range of yaw angle[deg]:	[-90, +90]
Actuators:	RC servo motor
	(Futaba S3305)
Sensors:	Pressure sensor (INASTOMER)
	IR sensor (SHARP GP2Y0A710K0F)

The mechanical structure of the joint unit is shown in Fig. 2.3. The original works of Hirose [19] have given out the mechanism for imitating snake-like locomotion. During the process of progressing, due to the particular scales and ribs of the snake, the friction coefficient of the contact in the normal direction with respect to the main axis of the segment is significantly greater than the tangential one. By installing a passive wheel on the snake-like robot, we can imitate these biological characteristics of the snakes in nature. The propelling force of the serpentine motion of a snake-like robot comes from the interaction between the passive wheel and the ground by swinging the joints from side to side. From

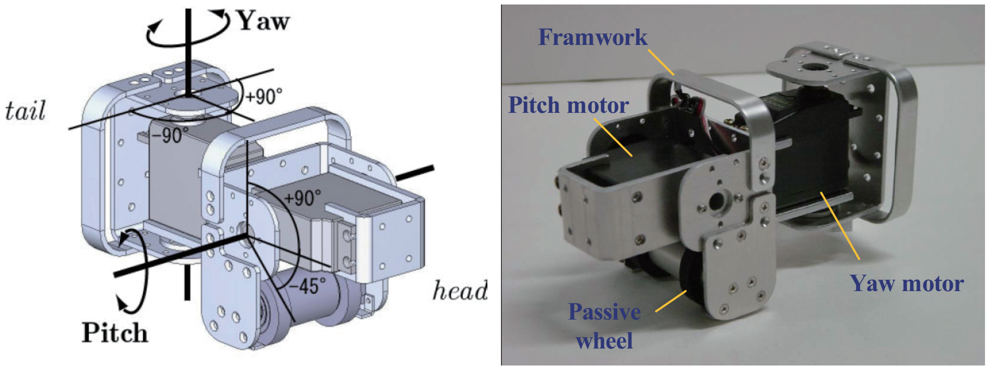
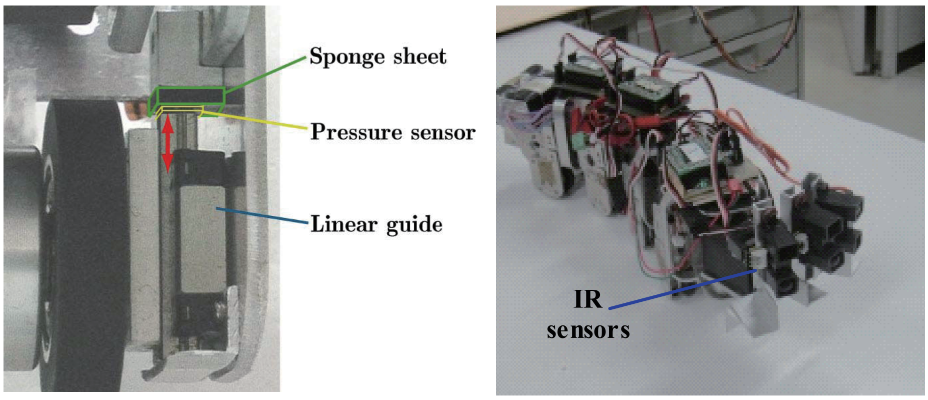


Figure 2.3. Structure of one link in *SR*-I

the Fig. 2.3, we can find that the link of the robot is composed of two motors and passive wheels which are fixed with a framework.



(a) (b)

Figure 2.4. Arrangement of the sensors in *SR*-I

To allow autonomously adapted motion of the snake-like robot according to the surroundings, the sensory information should be coupled into the robot system. Herein two kinds of sensors have been installed on the snake-like robot as peripheral receptors. One is the pressure conductive sensor which is used to detect ground information. The setting of the pressure sensor is shown in Fig. 2.4 (a). It is mounted on each wheel of the snake robot. By the using of a linear guide the contact force between the wheel and the ground can be detected through the displacement between the wheels and sensors. Another one is the infrared(IR) range sensor which is used to get surrounding obstacle information. The arrangement of the sensors is shown in Fig. 2.4 (b). Three IR sensors are installed on the head of the snake robot with certain angle to get the obstacle information in front of the locomotion.

2.2.2 Control System

The creeping motion of a real snake is generated by the rhythmic swing of the vertebrae. For the snake-like robot, one joint corresponds to one vertebra. Each joint needs an oscillator to imitate the muscle extension and contraction during the rhythmic swing of the body. For the control system of the snake robot, an oscillation network is constructed to generate a series of successive rhythmic signals with a certain phase difference. This control system should allow us to achieve coordination between large numbers of joints in the robot. When the rhythmic signals implement on the joint motors as angle values, the robot can perform desired serpentine motions.

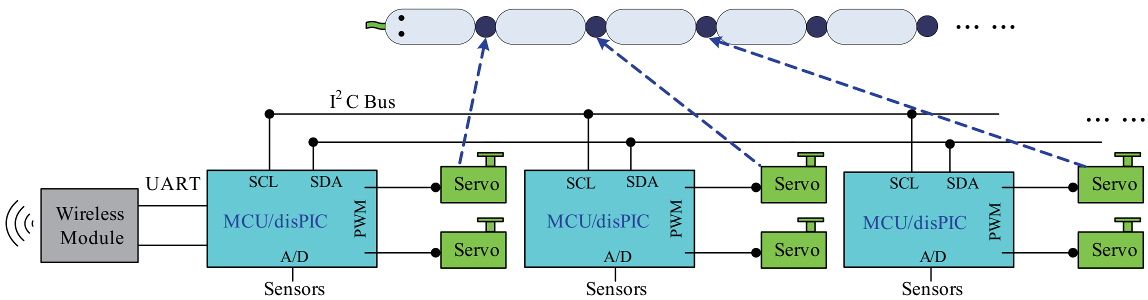


Figure 2.5. Structure of control system

The design of of the control system in the snake-like robot is shown in Fig. 2.5. One module corresponds to one driving motor at each joint. The motor controller is based on a dsPIC microcontroller (dsPIC33FJ64GP202), which drives the angle of the RC servo motor (FUTABA S3305) directly by the PWM signals. Furthermore, Inter-Integrated Circuit (I2C) bus has been adopted to realize the communication of the oscillation network. The direction of signal transmission is based on the ID address of each Micro-control Unit (MCU). The flow chart of processing procedure in the MCU is shown in Fig. 2.6.

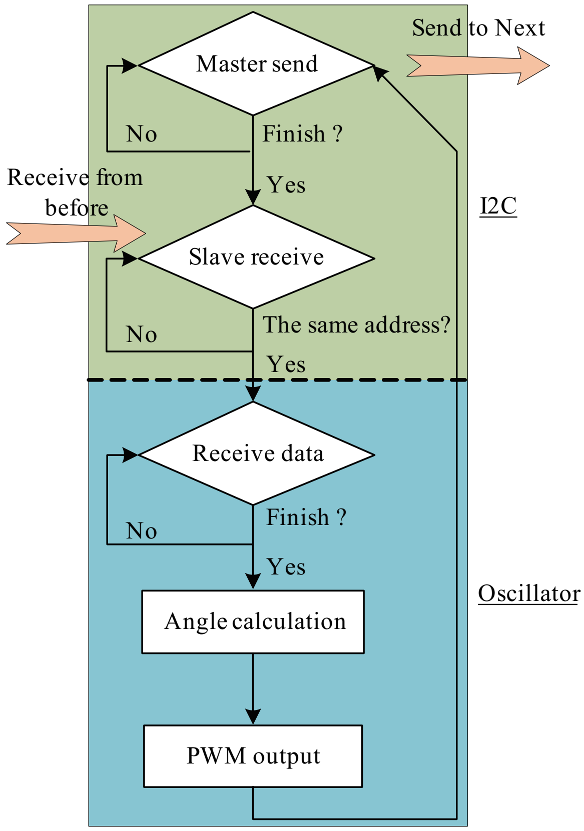


Figure 2.6. Flow chart of communication

The sensing equipments described in mechanical design are connected with MCU with A/D port. Through signal processing, the environmental information will be coupled into control system. Furthermore, a wireless module (XBee-PRO Module) using 2.4GHz band is installed on the head module to transmit or receive signals with external device, like wireless controller. This can also allow us to monitor the sensory information online or send high-level command by PC.

2.3 Summary

In this chapter, we introduced the design of the simulation and experiment platform. The motion of the snake-like robot can be simulated in an ODE-based dynamic environment. An sensor-driven snake-like robot *SR*-I is designed to perform desired serpentine locomotion in actual environment. In the following description, the proposed control system and motion patterns will be verified through the simulator and the physical experiment of SR-I.

Chapter 3

CPG Model and Network

The ability to efficiently move in complex environments is a key property of animals. It is central to their survival, i.e. to avoid predators, to look for food, and to find mates for reproduction. This capital property of animals related to locomotor skills have been shaped by nervous systems of animals. Similarly, providing good locomotor skills to robots is of primary importance in the design of the robots that can carry out useful tasks in various environments. This relationship of locomotion in biology and robotics has led to multiple interesting interactions between the two fields. Many robot structures are directly inspired by animal morphologies, from snake robots, quadruped robots, to humanoid robots. Increasingly, robotics is now providing something back to biology, with robots being used as scientific tools to test biological hypotheses [21] [12].

One of the biological hypotheses is that researcher found that the rhythmic motion patterns of the vertebrate animals is generated from a neural oscillator system, which is called central pattern generator (CPG). CPG-based oscillator models are based on mathematical models of coupled nonlinear oscillators to study population dynamics. Depending on the previous study, CPG models have been designed at several levels of abstraction from detailed biophysical models, to connectionist models or coupled oscillators. In some cases, the CPG models have been coupled to biomechanical simulation where they are called neuromechanical models [21].

The purpose of these models is not to explain rhythm genesis but to study how inter-oscillator couplings and differences of intrinsic frequencies affect the synchronization and the phase lags within a population of oscillators. The motivation of modeling comes from the fact that the dynamics of populations of oscillators depends mainly on the topology of network rather than on the local mechanisms of rhythmic generation [14] [31]. See for instance the study by Collins and Richmond [6] which obtain the same gait transitions in a given network topology with three different types of oscillators (van der Pol, Stein, and FitzHugh-Nagumo). Other extensively used oscillators include phase oscillators [5] [56] [30] and Matsuoka oscillators [40] [64] [28]. Several neuromechanical models have been developed in [69] [64] [10] [23] [22]. The interaction of a CPG model with the environment offers the possibility to study the effect of sensory feedback on the CPG activity. Important phenomena such as mechanical entrainment can thus be studied.

Most of the oscillators in previous studies are designed in a biological perspective which are structurally complicated and not suitable for purposive control or numerical analysis in engineering viewpoint [73]. In this paper, based on the neural model of Matsuoka [40] [41], we developed a CPG oscillator network to control our snake-like robot. The model of CPG neuron proposed by Matsuoka has the features of continuous-time and continuous-variable in its simple structure, and can thus be easily implemented onto the control of the snake-like robot with large DOFs. Moreover, the Matsuoka's CPG model has been taken stable analysis and mathematical proof to generate rhythmic output in our previous work [33]. In this chapter, the design of the CPG-based neural controller will be described in detail.

3.1 Central Pattern Generator

In the design of the CPG model, rhythmic oscillator requires: (1) two or more neurons that interact such that each neurons sequentially increases and decreases, and (2) that, as a result of this interaction, the system repeatedly returns to its starting condition. In Matsuoka's model, one CPG can be composed of several neurons corresponding to extensor and flexor muscles. Each neuron has membrance potential (internal state) and fatigue effect

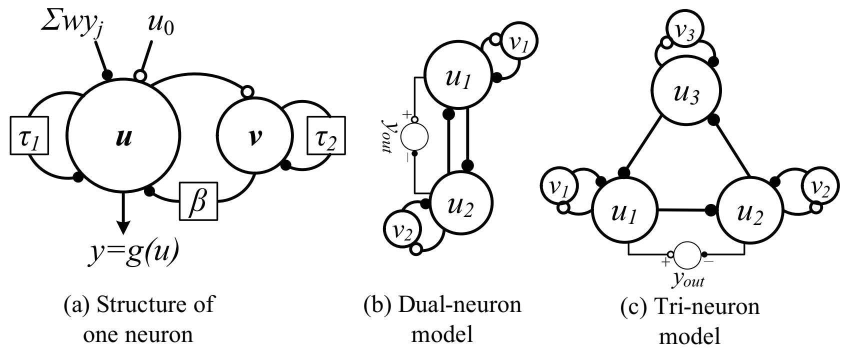


Figure 3.1. CPG models of neural oscillator

(self-inhibition). These neurons have mutual inhibitory connection. Besides, neurons receive tonic driving inputs from upper center, which drives CPG oscillation. Neurons receive input from other CPGs and sensory feedback signal, which coordinates CPG dynamics.

3.1.1 Neural Model

The structure of the individual neuron model is shown in Fig. 3.1 (a). The mathematical model of each neuron can be expressed as

$$\begin{aligned} &\tau_1 \dot{u} + u = u_0 - \beta v - \sum_{j=1}^{m} w y_j \\ &\tau_2 \dot{v} + v = y \\ &y = g(u) = max(0, u) \end{aligned} \tag{3.1}$$

where u is the membrane potentials of the neuron; v is the variable that represents the degree of adaptation; y is the output of the CPG neuron, and its value is always positive; u_0 is the tonic driving input; τ_1 and τ_2 are the parameters that specify the time constants for membrane potential and adaptation degree, respectively; β is the adaptation coefficient; w is the weight between neurons; $\sum w y_j$ represents the input from other neurons; m is the number of all the neurons in one CPG model.

Usually, there are several ways to construct a CPG model by connecting different number of neurons. For instance, a dual-neuron model and a tri-neuron CPG model are given in Fig. 3.1 (b) and Fig. 3.1 (c), respectively. These two CPG models have already been adopted in the control systems of many bionic robots [29] [33]. Due to the complicated structure and numerous computations, the CPG model with four or more neurons is little used in practical applications.

Since the output of an individual neuron is always positive value, the output of the CPG module y_{out}, which is defined by the subtraction between the output of the first neuron y_1 and that of the second neuron y_2 in the CPG module, is used to get a symmetrical rhythmic signal with both positive and negative values.

$$y_{out} = y_1 - y_2 \tag{3.2}$$

For one CPG model, an initial stimulation is needed to promote the neurons to generate a group of rhythmic signals with certain phase, like a high level command from cerebrum in the biological CPG.

3.1.2 Comparison of Two CPG Models

In a previous research, a mutual inhibition model was successfully adopted for a snake-like robot to realize a meandering locomotion [24]. It has also been proven that a cyclic inhibitory CPG model with triple neurons is more suitable for a snake-like robot control [33]. Two outputs can be provided by this cyclic inhibitory CPG model to control the yaw and pitch movement of the snake-like robot, respectively. Furthermore, this cyclic inhibitory CPG model does not necessitate adaptation neuron, and its internal strong cyclic inhibition can perform rhythm generation. Moreover, it has been proven that this model can realize a 3D locomotion with the least computation cost in the neural control system [34].

By using the parameters setting in Table. 3.1, the output y_{out} of the mutual inhibitory model and that of the cyclic inhibitory model are obtained, as shown in Fig. 3.2. It can be easily found that the output of the mutual inhibitory model has more flex points than that of the cyclic inhibitory model. Since the output y_{out} is the input signal to control the angle of joint in the snake-like robot, the derivative of output dy_{out}/dt has a direct effect on the actuator of the joint. It can be observed that the curve of dy_{out}/dt in the cyclic inhibitory model is smoother than that in mutual inhibitory model. Therefore, the joint can perform

Table 3.1. Parameters of the CPG model

Driving input u_0	2.5
Time constant τ_1	2.0
Time constant τ_2	6.0
Adaptation coefficient β	2.5
Connection weight inner neurons w	2.7
Connection weight among CPGs w_0	0.1

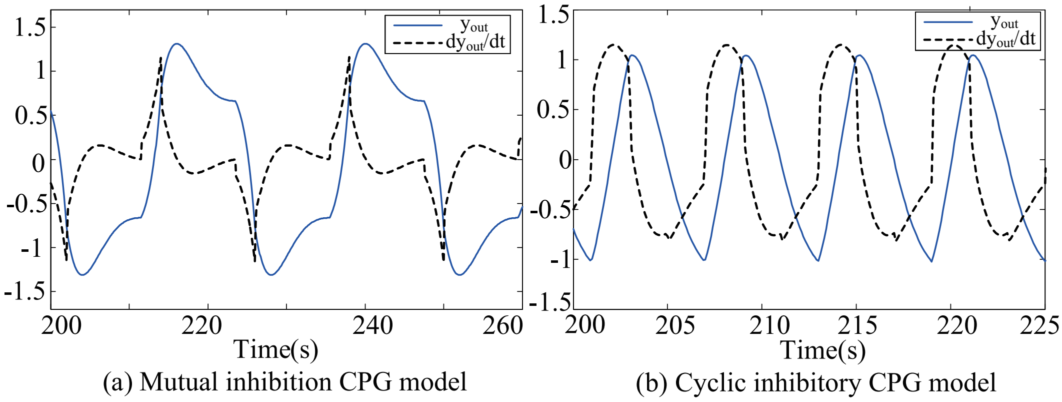


Figure 3.2. Outputs of two CPG models and their derivatives

a better motion when a cyclic inhibitory model is adopted. The cyclic inhibitory model is thus selected as the CPG network model to control our snake-like robot.

3.2 CPG Network with Feedback Connection

The propelling force of the serpentine motion of a snake-like robot comes from the interaction of the robot with the ground just by swinging the joints from side to side [19]. The rhythmic signals implemented on the joint motors, can be easily generated by a CPG network like Fig. 3.3. Due to the fact that one joint angle corresponds to one CPG output, a series of successive rhythmic signals with certain phase difference are needed to realize snake-like locomotion control. Thus, several CPG modules are needed to construct a kind of network for mimicking the neural system of a natural snake.

For simplicity, an open-loop unilaterally connected CPG network has been efficiently

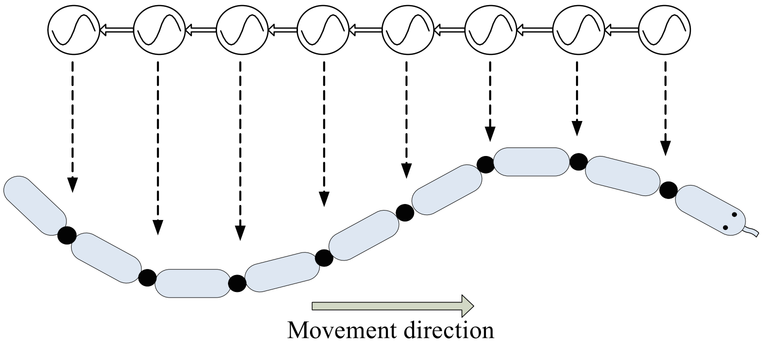


Figure 3.3. CPG network implemented to control a snake-like robot

employed for the control of the snake-like robot in [24] and [33]. However, additional calculation is needed to adjust the irregular output signal of this network. To solve the problems in the unilateral CPG network, a closed-loop network with unidirectional couplings between the oscillators is proposed in this paper. Similarly, a cross connected network in [39] also has a closed-loop architecture. However, the number of connections in this network is two times more than our network and the calculation is increased correspondingly. Furthermore, the phase difference in this network can not be changed conveniently because of the strong couplings between the CPGs.

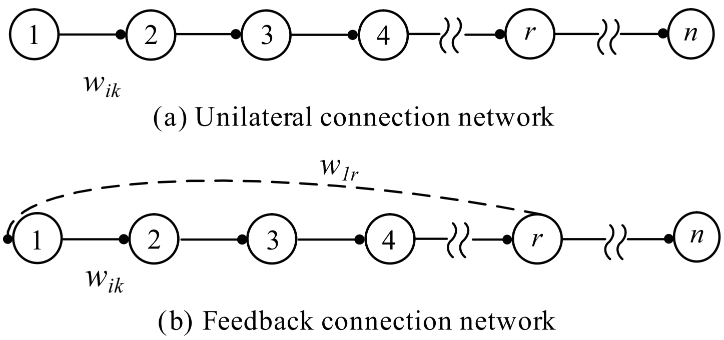


Figure 3.4. Schematics of different CPG connection networks

3.2.1 Mathematical Model of CPG Network

Based on the mathematical model of a single neuron stated in section 2, a CPG network includes n CPG modules which have m neurons can be described in a group of basic equations. Regarding the j-th neuron of the i-th CPG module, the mathematical model can be described by

$$\begin{aligned}
&\tau_1 \dot{u}_{j,i} + u_{j,i} = u_0 - \beta v_{j,i} - w y_{s,i} + \sum_{k=1}^{n} w_{ik} y_{j,k} \\
&\tau_2 \dot{v}_{j,i} + v_{j,i} = y_{j,i} \\
&y_{j,i} = g(u_{j,i}) = max(0, u_{j,i}) \\
&y_{out,i} = y_{1,i} - y_{2,i} \\
&i, k = 1, 2, ...n,\ i \neq k; j = 1, 2, ...m; \\
&s = \begin{cases} m, & \text{if } j = 1 \\ j - 1, & \text{others} \end{cases}
\end{aligned} \tag{3.3}$$

where n is the number of CPG modules in the network; m is the number of neurons in one CPG module; s is the serial number of neuron connected to the j-th neuron; $u_{j,i}$ is the membrane potentials of j-th neuron in the i-th CPG module; $v_{j,i}$ is the variable that represents the degree of adaptation; u_0 is the tonic driving input; τ_1 and τ_2 are the time constants; β is the adaptation coefficient; w is the weight between neurons; w_{ik} is the connection weight of the i-th module from the k-th module; $y_{j,i}$ is the output of j-th neuron in i-th CPG module; $y_{out,i}$ is the output of the i-th CPG module.

3.2.2 Unilaterally-connected Network

The open-loop unilaterally-connected network was employed for the snake-like robot control in previous research, as shown in Fig. 3.4 (a). Herein, each circle represents one CPG module, which can be a dual-neuron mutual inhibition model or a tri-neuron cyclic inhibitory model. In this network, all the CPG modules are connected in the direction, from the head to the tail, in which there is a constant connection weight between the neighboring

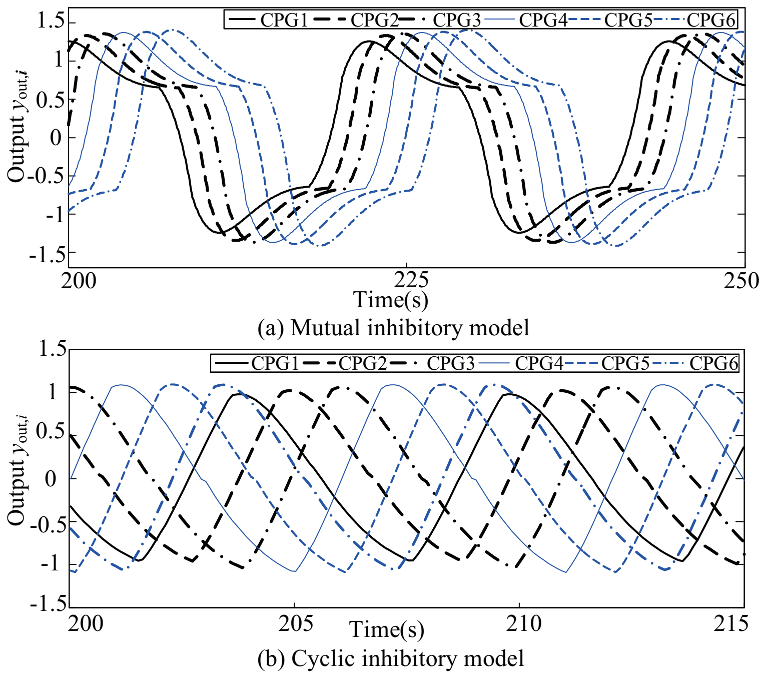


Figure 3.5. Rhythmic outputs of two kinds of CPG models using unilaterally-connected network

modules.

$$w_{ik} = \begin{cases} w_0, & k = i - 1 \\ 0, & others \end{cases} \tag{3.4}$$

A unilaterally-connected network with six CPG modules is simulated by use of the parameters setting in table 5.1. Rhythmic outputs of both the dual-neuron mutual inhibition model and the tri-neuron cyclic inhibitory model are given in Fig. 3.5 (a) and (b).

3.2.3 Closed-loop Network with Feedback Connection

The basic concept of this network is shown in Fig. 3.4 (b). Compared with the unilaterally-connected network, this network has the same unidirectional couplings between the oscillators. The difference is that the output of the r-th CPG module is provided as feedback to the first module, so as to form a closed loop. The connection weight between CPGs w_{ik} takes the same value from equation (3.4). Besides, the weight of the feedback w_{1r} from the r-th module to the first one should also adopt a weight value, given by $w_{1r} = w_0$.

Since the r-th CPG module transmits the same value to the (r+1)-th and the 1-st CPG module, the output waves of the (r+1)-th and the 1-st module have completely identical shape and phase. The (r+1)-th module will also transmit its output to the (r+2)-th module as the 1-st module transmits its output to the 2-nd module. Thus, the (r+2)-th module will generate the same output as that of the 2-nd module. In the same way, the following output of CPG modules will take the same manner as CPGs in the loop. Thus, the output of the CPG modules on the outside of the loop can be represented as

$$y_{out,pr+q} = y_{out,q} \quad p = 1, 2..., \quad q = 1, 2...r \tag{3.5}$$

A network with six CPG modules, where the 5-th module is selected to provide the feedback to the first CPG module, is simulated by use of both the dual-neuron mutual inhibition model and the tri-neuron cyclic inhibitory model. Rhythmic outputs of CPG models with respect to the same set of CPG parameters in table 5.1 are shown in Fig. 3.6 (a) and (b). From the figures, we can find the signal of the 6-th CPG is coincident with that of the first CPG. This relation of $y_{out,6} = y_{out,1}$ can also be confirmed by the equation (3.5) where $r = 5$,$p, q = 1$.

There is an interesting characteristic in this closed-loop connection network. It is obvious that all the output waves of the CPG modules in the loop are homogeneously distributed in one period, as shown in Fig. 3.6. Since the output of the CPG modules on the outside of the loop has the same phase as CPGs in the loop, the phase difference between the two neighboring CPG modules in the whole network can be obtained by

$$\Phi = 2\pi/r \tag{3.6}$$

3.2.4 Comparison of Two CPG Networks

As shown in Fig. 3.5, the amplitude of the CPG outputs in the unilaterally-connected network is not of uniform size and the waveforms can not form a perfect traveling wave to suit the control of the multi-link robot. Thus it is necessary to make adjustments to get applicable rhythmic signals for the control of the robot. Furthermore, it is also difficult to adjust one or more parameters to obtain the desired phase difference of outputs.

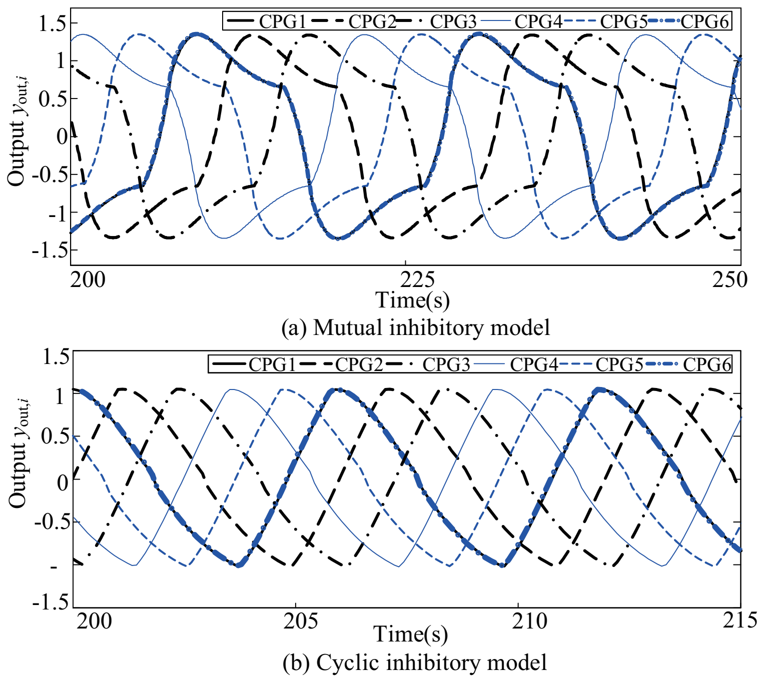


Figure 3.6. Rhythmic outputs of two kinds of CPG models in closed-loop network

Compared with the unilaterally-connection network, the closed-loop network with feedback connection, which employs the two CPG models, generates more uniform outputs with the same amplitude and phase difference as in Fig. 3.6. Due to the fact that this network can be applied to snake-like robot control without any additional modification on the output signals of the CPG, the system computation is decreased dramatically. The calculations for the adjustment of the phase and amplitude in [34] are unnecessary. Each oscillator has been reduced two equations with four variables in the CPG model. Thus for the snake robot with a group of CPGs, multiple numbers of equations will be eliminated. Moreover, from the above discussion, we know that it is more convenient to obtain a specific phase difference by feeding one of the CPG signals back to the first module. This characteristic can be used to control the number of S-shape waves in snake-like robot locomotion, as stated in section 4.2. Therefore, this network with the feedback connection is more suitable for the control of a snake-like robot.

3.3 Characteristic of CPG network

3.3.1 CPG Parameters

To figure out how to employ the CPG network to control locomotion of a snake-like robot, the influence of each CPG parameter on the signal output has to be investigated. In Matsuoka's early work [40] [41], some partial and qualitative conclusions on the influence of the parameters onto the oscillator behavior have been indicated. But it has not given a detail analysis of the CPG characteristics to generate the desired rhythmic signals in the network. Since there is a strong coupling relation between the parameters and the output, herein a numerical method is used to get the characteristics of the CPG model. To find out the influence of each parameter on the output, the output characteristic is studied with value of one parameter varied in a certain range while the rest of the parameters remain unchanged.

From the mathematical model of the CPG model in equation (5.13), the output is mainly determined by several parameters. Influence of these parameters on the amplitude and the period of oscillator output is primarily investigated. A cyclic inhibitory CPG model with three neurons is selected to study the influence of parameters, and the numerical results are shown in Fig. 3.7. The value ranges of these parameters conform to the mathematical analysis for stable oscillation in [41], where every parameter should meet the following numerical condition:

$$w/(1+\beta) \leq 1 \quad and \quad 1+\tau_1/\tau_2 < w \tag{3.7}$$

Summarizing these characteristics, a concise conclusion can be obtained in table 3.2. Two important linear relations can be easily found: output amplitude increases linearly with the driving input u_0; time constant τ_1, τ_2 keeps a linear relation with the period of output while the value of τ_1/τ_2 is a constant. However, if τ_1/τ_2 is not a constant, the linear relation between time constant τ_1, τ_2, and the period of output will be broken. The change of parameter u_0 does not influence the period of the output while the change of τ_1, τ_2 makes no influence on the amplitude of the output. Furthermore, the basic shape of the output

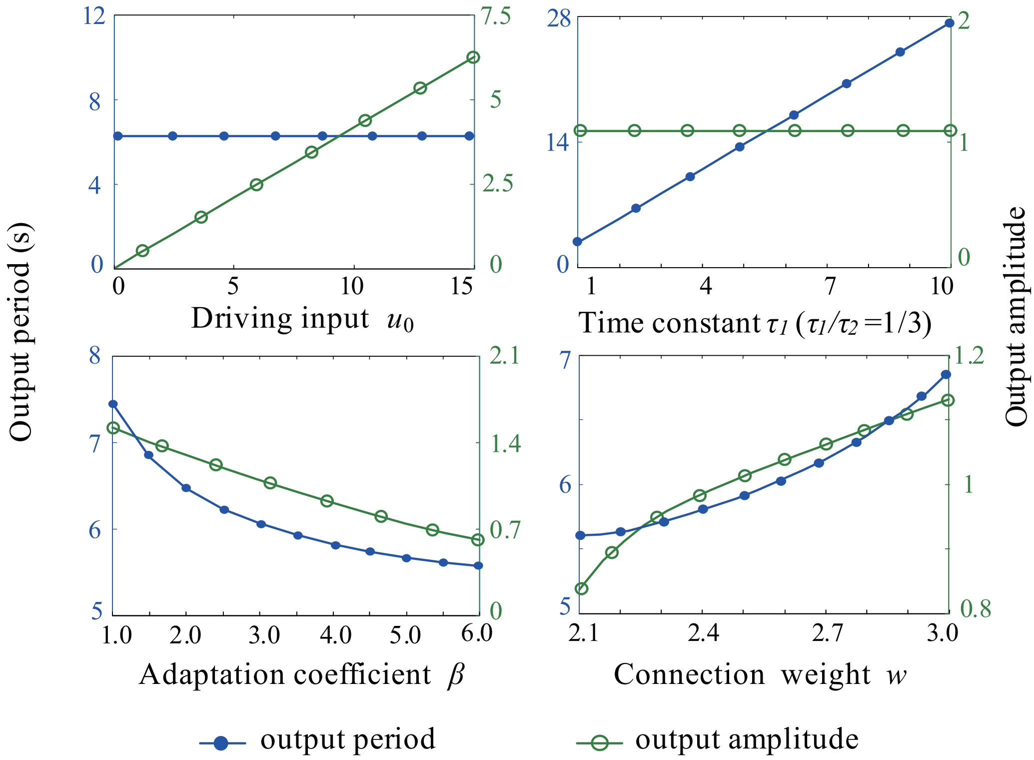


Figure 3.7. Results to show the relation between the CPG parameters and the CPG output

wave is also not affected by the driving input and time constant. For the parameters of the coefficient β and connection weight w, there are no these advantages. Thus, the driving input and time constant can be employed to adjust the CPG output by these two useful linear relations conveniently. Fig. 3.8 shows the oscillation signal changed with respect to driving input and time constant respectively.

3.3.2 Number of the Locomotion S-shape

A snake-like robot performs locomotion with one S-shape when the sum of the total phase differences of the joints is 2π. Number of the locomotion S-shape increases with respect to the increasing phase difference of joint angles. As stated in section 3, the phase differences of CPG outputs are homogeneously distributed in one period if the network is a closed-loop type. From equation (3.6), the sum of the total phase differences of CPG's outputs can be derived by $(n-1)\Phi$ where n is the numbers of CPGs. Since one S-shape

Table 3.2. Change of the CPG output with respect to each CPG parameter

Parameters increase	Output Amplitude	Output Period	Wave shape of output
u_0	linear increase	unchangeable	unchangeable
τ_1, τ_2 with constant τ_1/τ_2	unchangeable	linear increase	unchangeable
β	nonlinear decrease	nonlinear decrease	unchangeable
w	nonlinear increase	nonlinear increase	changed

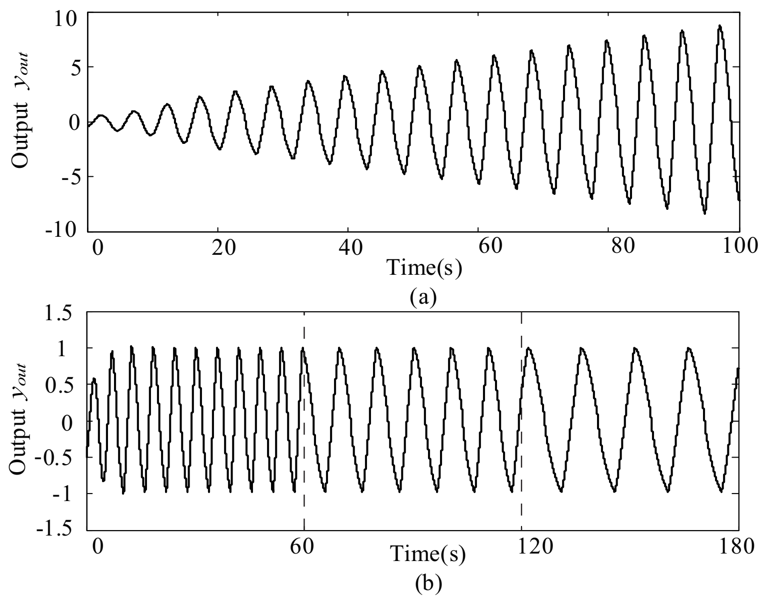


Figure 3.8. Oscillation signals changed with respect to Driving input and Time constant where (a) Driving input $u_0 = t/5 + 2.5$; (b) Time constant τ_1=2.0, 3.5, 5.0 and τ_2=6.0, 7.5, 15.0 for time t=0, 60, 120 (s)

locomotion can be obtained by a group of rhythmic signals with total 2π, number of the locomotion S-shape, N, can be given by

$$N = \frac{(n-1)\Phi}{2\pi} = \frac{n-1}{r} \tag{3.8}$$

Therefore, the number of the locomotion S-shape can be varied by changing the connection of r-th CPG module to the first CPG module.

3.4 Summary

In this chapter, a bio-inspired system imitating the CPG neural network has been proposed as the control method of a snake-like robot. The CPG network with a feedback connection does not necessitate to take additional adjustments of the CPG output due to its uniform outputs with the same amplitude and specific phase difference. The influences of CPG parameters on the rhythmic output were investigated and the relation curves were obtained. Based on these results, the following CPG-based motion control can be realized.

Chapter 4

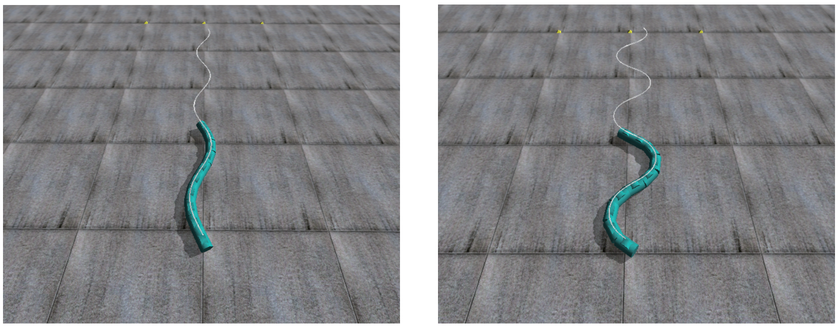
CPG-based Motion Control

To figure out how to employ the CPG network to control locomotion of a snake-like robot, the influence of each CPG parameter on the signal output has to be investigated. In our description in chapter 3, two important linear relations between the CPG parameters and the rhythmic output has been found: output amplitude increases linearly with the driving input u_0; time constant τ_1, τ_2 keeps a linear relation with the period of output while the value of τ_1/τ_2 is a constant. Furthermore, the change of parameter u_0 does not influence the period of the output while the change of τ_1, τ_2 makes no influence on the amplitude of the output. The amplitude and the period of the joint angle influence the motion curvature and motion frequency of the snake robot respectively. Based on these results, CPG-based motion control of a snake-like robot can be realized by adjusting these parameters.

4.1 Basic Motion Patterns

4.1.1 Curvature of Locomotion

A natural snake often changes the curvature of its body to adapt to different terrains during locomotion. For instance, a large locomotion curvature to adapt to a slippy ground. In the simulation, we find that if the amplitude of CPG output increases, the curvature of the snake-like robot will increase correspondingly. Thus, due to the linear relation between the CPG output amplitude and the parameter driving input u_0, a different locomotion curvature can be obtained by adjusting u_0. From the simulation results shown in Fig. 4.1, we know that the curvature of the robot becomes larger when the driving input u_0 increases from 1 to 5.5. Fig. 4.2 shows the average curvature, which is derived from the reciprocal value of the snake-like robot axial distance $1/l_{axis}$ (l_{axis} is the distance of the snake-like robot along its axial axis), changed with respect to increased driving input. Here we take one S-shape for locomotion and other parameters are fixed in the value given in table 5.1.



(a) Locomotion trajectory when u_0=1.0

(b) Locomotion trajectory when u_0=5.5

Figure 4.1. Locomotion of the robot with different body curvatures

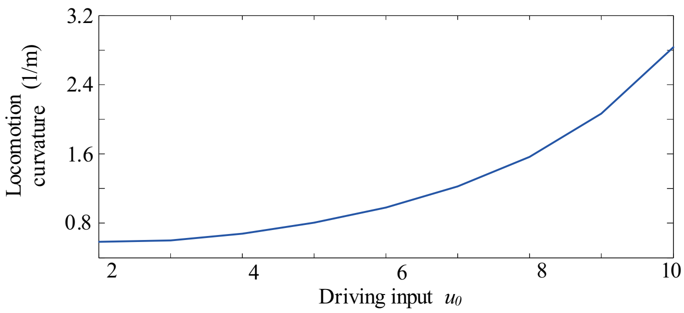


Figure 4.2. Curvature with respect to CPG driving input

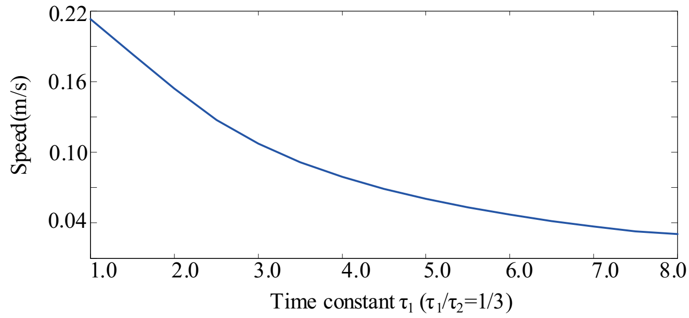


Figure 4.3. Motion speed with respect to CPG time constant

4.1.2 Speed of Locomotion

Robot speed is mainly affected by locomotion frequency. As stated above, the period of CPG output keeps a linear relationship with time constant. Thus, the speed of the robot can be controlled by proportionally adjusting the time constant. Herein, time constant τ_1 and τ_2 are adjusted with the value of τ_1/τ_2 kept at a constant. From the result of the simulation shown in Fig. 4.3, we can find that the motion speed decreases with the increasing of the time constant, since this increase makes the period of CPG output become longer.

4.1.3 S-shape of Locomotion

The number of S-shape is an important motion parameter in snake-like locomotion, which describes the number of periods of the sinusoidal wave existing in snake-like locomotion shape. Based on the above analysis, we change the CPG network by different feedback connections to obtain the desired number of locomotion S-shape. For a snake-like robot with 10 joints in the simulation, when connecting the first CPG module with the 9-th, the robot can generate only one S-shape, shown in Fig. 4.4 (a); when connecting the first CPG module with the 5-th, it can generate 1.8 S-shape, shown in Fig. 4.4 (b). The same results can also be obtained from the equation (3.8) where $n = 10$, $r = 9$ and 5 respectively.

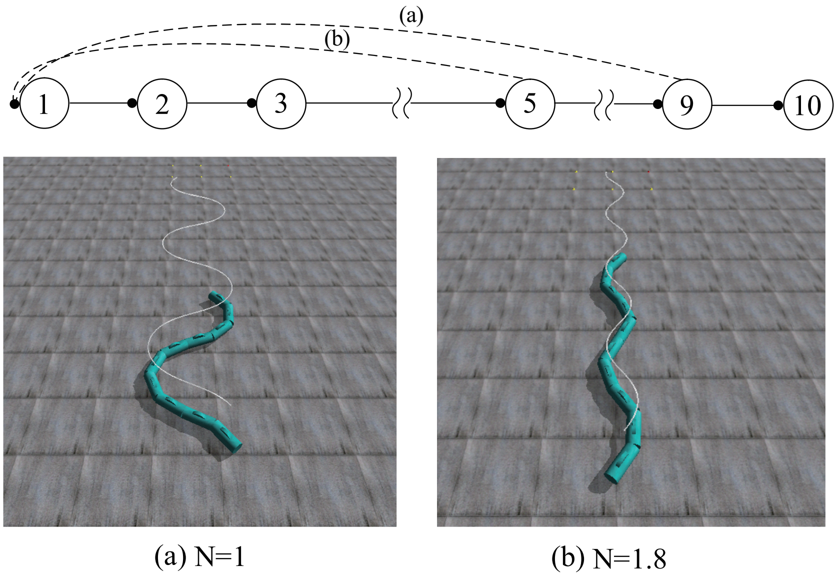


Figure 4.4. Numbers of S-shape changed through feeding a different CPG back to the first one

4.2 Motion Optimization

In the previous section, we have verified through simulations that our CPG network is capable of achieving kinds of locomotion by the adjustment of parameters. In the nature the snakes can always keep an adaptive body shape to move on kinds of terrains. In order to figure out which parameters can generate an adaptive motion corresponding to different environments by the CPG-based control, the creep locomotion with the experimental optimization is discussed through simulated experiment. Herein, two typical environments, including slope and horizontal ground with different frictions, are considered to obtain the efficient locomotion of the snake-like robot.

4.2.1 Curvature Adaptive Principle

The Coulomb friction model is adopted to describe the interaction between snake-like robot and ground. Detailed analysis about this robot dynamics can be seen in [37]. The tangential and normal frictions exerted on the $p^{th}(p = 1, 2...)$ module of snake-like robot can be expressed by

$$
\begin{aligned}
f_p^t &= -\mu_t m_p g \cdot \cos\phi \cdot sign(\delta^p r^t) \\
f_p^n &= -\mu_n m_p g \cdot \cos\phi \cdot sign(\delta^p r^n)
\end{aligned}
\tag{4.1}
$$

where f_p^t and f_p^n are the components of friction in tangential and normal directions, respectively; μ_t and μ_n are the relevant friction coefficients; m_p is the weight of the p^{th} module, which includes both the mass of the link and that of the wheel; ϕ is the inclined angle of slope; $\delta^p r^t$ and $\delta^p r^n$ are the tangential and normal displacements of the p^{th} module at friction point, respectively.

By setting the coordinate system shown in Fig. 4.5, where the x-axis is taken along or parallel to the forward direction of snake-like robot, the x and y components of the resultant friction force can be calculated by

$$
\begin{aligned}
f_p^x &= f_p^n \sin\psi_p - f_p^t \cos\psi_p \\
f_p^y &= f_p^n \cos\psi_p + f_p^t \sin\psi_p \\
F_x &= \sum_{p=1}^{N} f_p^x \\
F_y &= \sum_{p=1}^{N} f_p^y
\end{aligned}
\tag{4.2}
$$

where f_p^x and f_p^y are the components of resultant force in x and y directions, respectively; ψ_p is the rotation angle from x-axis to each link. By taking the snake-like robot as a whole, F_x and F_y are the resultant propellent force from the friction of each link along x-axis and y-axis, respectively.

To provide propellent force of the snake-like robot along x-axis, it should have a enough friction to avoid backward slippage. Combining the equation (4.1) and equation (4.2), a positive value of F_x along the forward direction of the robot can be got by an asymmetric friction with μ_n larger than μ_t. However, when the difference between two friction coefficients is not big enough to obtain sufficient F_x, the snake-like robot will swing without effective forward motion. The increment of locomotion curvature with increased ψ_p can enlarge f_p^x to get suitable F_x and solve this problem. From the description of locomotion

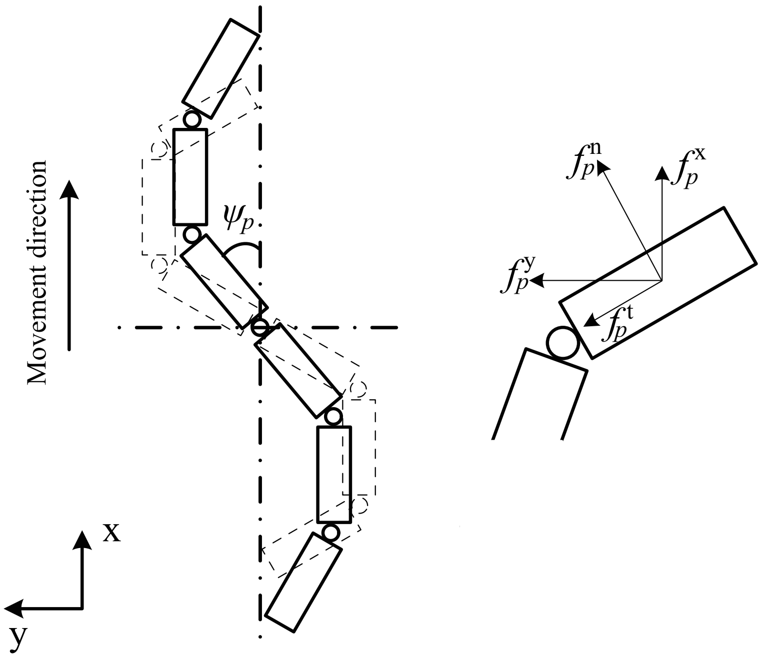


Figure 4.5. Curvature adaptive principle

curvature in section 4, this curvature adaptive principle can be conducted easily by the CPG-based control.

Despite a large locomotion curvature can avoid the skid on the ground, the movement speed along the forward direction will decrease correspondingly. Besides, the force used to drive the robot is also influenced by the locomotion curvature. In order to evaluate the locomotion efficiency of the snake-like robot under different locomotion curvature, we propose a criterion of optimal locomotion, where the ratio between the forward displacement and energy consumption is considered as the target of evaluation. This coefficient of locomotion efficiency can be described as

$$J = \frac{S_T}{\int_0^T \sum_1^n tor_i^2 dt} \tag{4.3}$$

where S_T is the total forward displacement along the longitudinal direction in one period; tor_i is the torque added on i^{th} joint, thus $\int_0^T \sum_1^n tor_i^2 dt$ is the total square-sum of joint torques in one period; n is the number of robot joints.

Based on the proposed criterion, the effective locomotion curvature can be realized by minimizing the energy consumption and maximizing the forward distance, thus obtaining high efficiency for creeping locomotion of the snake-like robot.

4.2.2 Ground with Different Frictions

The normal and tangential frictions play an important role during the creeping locomotion of snake-like robot. As stated above, when the friction changes with respect to different textures of contact surface, the locomotion curvature can be adjusted correctly to get enough propulsion for the robot. Thus, the relevant parameters of CPG network should be modified to get the adaptive motion curvature. As stated in section 4, the driving input u_0 can be used to obtain the desired locomotion curvature of robot.

In the simulation, the interaction between robot and ground is modeled with variable friction environment, where the coefficient of normal friction is set as μ_n =0.5, and the coefficient of tangential friction μ_t varies from 0.01 to 0.25. Meanwhile, the parameter u_0 is adjusted from 1 to 15 to find the optimal locomotion curvature of the snake-like robot.

The motion displacement along the forward direction and the total square-sum of joint torques in one period, with respect to different friction coefficients and CPG driving input u_0, are illustrated in Fig. 4.6 (a) and (b). Due to the snake-like robot can only swing in the original place with S_T=0 in the blue region of Fig. 4.6 (a), it has a relevant limit inferior for the driving input u_0 to realize forward motion of the snake-like robot under different friction environment. The energy consumption of robot holds a nonlinear U-shape curve with the driving input u_0 in Fig. 4.6 (b). Based on the above proposed criterion, the locomotion efficiency of the snake-like robot with respect to the friction coefficient and CPG driving input is obtained in Fig. 4.6 (c), where the bottom coordinate of illustration is rotated to get a better view of the 3-D surface. For each μ_t, there is a corresponding u_0 to make the robot get the highest efficiency. Therefore, the value of drive input u_0 to generate the highest locomotion efficiency is optimized as shown in Fig. 4.7, while the coefficient of tangential friction μ_t varies from 0.01 to 0.25 with the interval of 0.02.

When implementing this optimization on the robot, by collecting the friction informa-

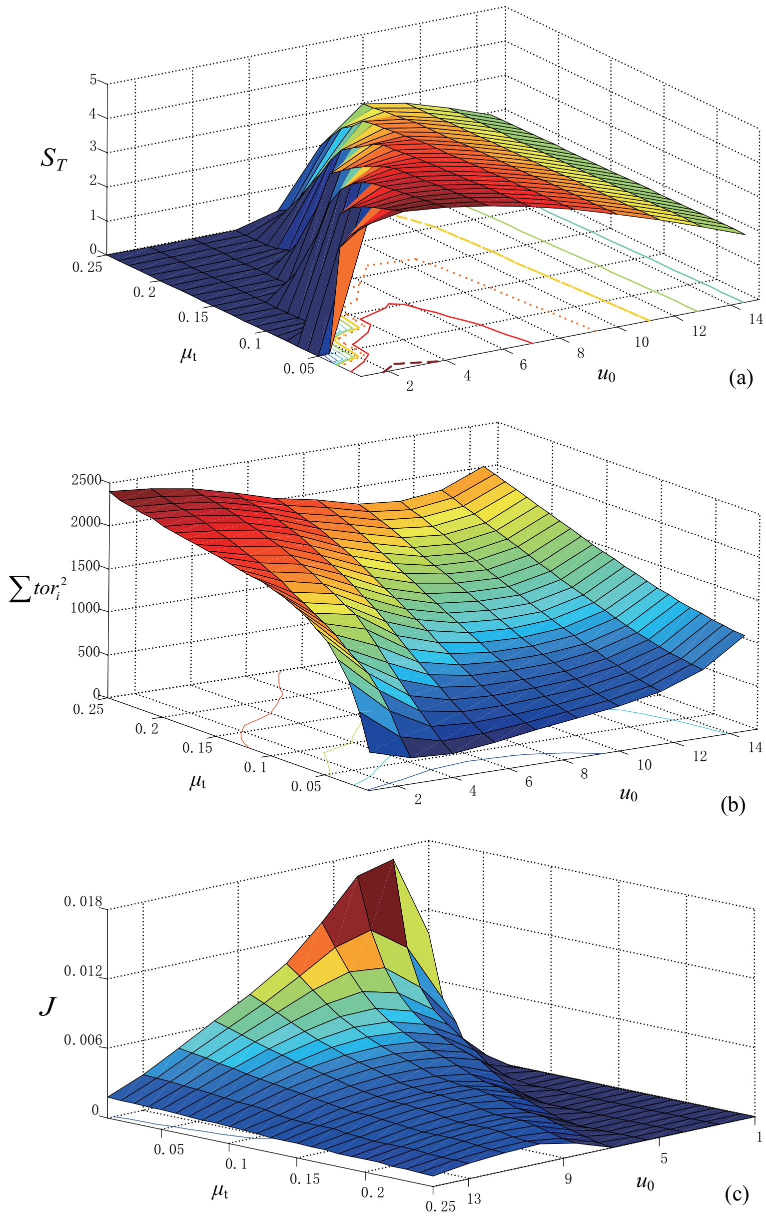


Figure 4.6. Simulation results under different friction

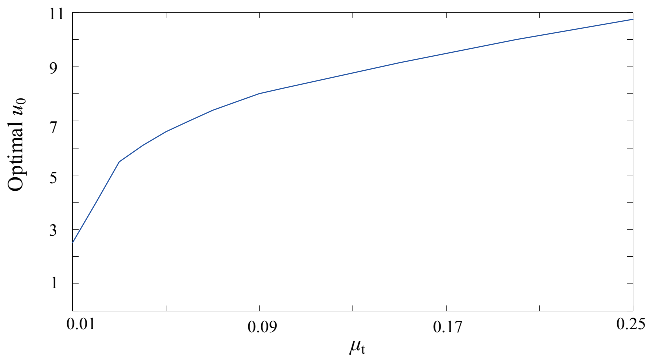


Figure 4.7. The relation between optimal u_0 and friction coefficient μ_t

tion of the ground from sensors, the value of the driving input u_0 can be adjusted to follow the data result like that in Fig. 4.7. Subsequently, the snake-like robot can creep with an adaptive locomotion curvature on the ground correspondingly.

4.2.3 Slope with Different Inclined Angles

When a snake-like robot creeps on a slope, it has to change its locomotion curvature to avoid slipping from the surface of slope. Thus, by adjusting CPG driving input, the snake-robot can creep on the slope with an adaptive locomotion curvature. As stated above, the locomotion on a slope can be optimized based on locomotive efficiency criterion in the same way.

In this simulation, the friction coefficients in tangential and normal directions are set as 0.03 and 0.5, respectively. The range of slope angle ϕ is from 0 to 10 degrees, and driving input u_0 is taken from 1 to 15. Fig. 4.8 illustrates the simulation results, which have the same trend as that in Fig. 4.6. It also has a limit inferior value of driving input u_0 to take forward motion of the snake-like robot on a certain slope from the blue region of Fig. 4.8 (a).

To get the highest locomotion efficiency for the robot, the relation between optimal driving input u_0 and incline angle ϕ is obtained, as shown in Fig. 4.9. By sensing the incline angle of the ground, the locomotion curvature of the snake-like robot can be optimized by

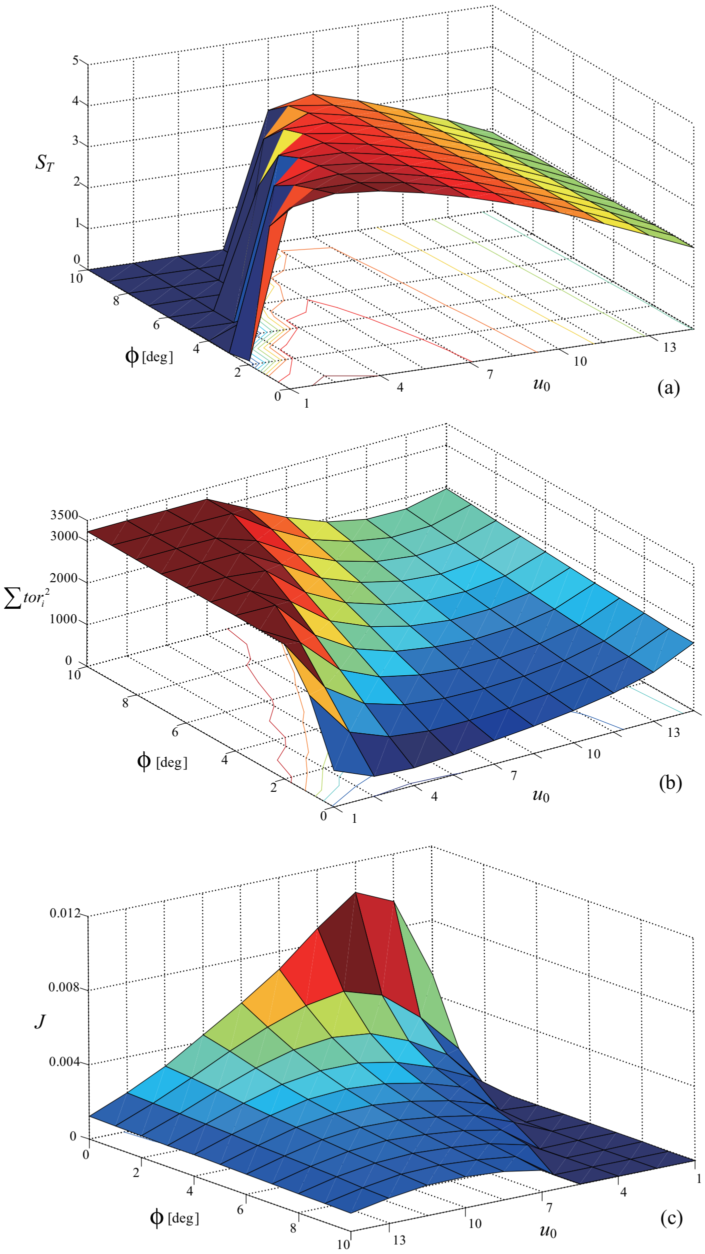


Figure 4.8. Simulation results under different incline angle

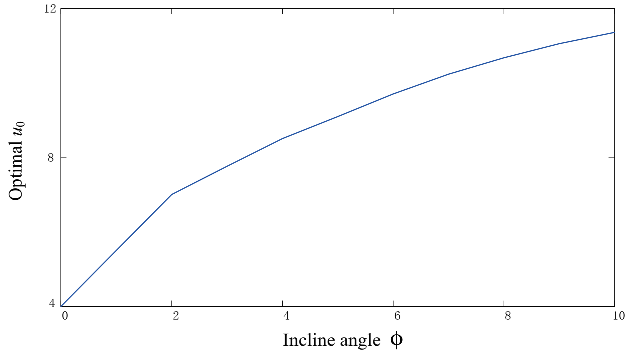


Figure 4.9. The relation between optimal u_0 and incline angle ϕ

adjusting the driving input u_0 accordingly as that in Fig. 4.9. Compared with the simulation result under different friction coefficients shown in Fig. 4.7, it can be found that both of two relation curves show similar tendency, since the slope is equivalent to the change of ground friction, which can be derived from equation (4.1).

An experiment of the creeping motion on the slope (about 10 degree) with different driving input has been conducted in Fig. 4.10. We found that the small value of driving input ($u_0 = 2$) can not drive the robot to climb up the slope. Only larger values (u_0=6 and 10) can realize the creeping motion on the slope. This has verified the proposed curvature adaptive principle of the snake robot.

4.2.4 Discussion of Ratio μ_n/μ_t

The simulation given in subsection 5.2 specifies the coefficient of normal friction μ_n as a constant value. To figure out the optimal locomotion when snake robot moves on the ground with different tangential and normal friction, μ_n/μ_t is considered as a variable. We find that the ratio μ_n/μ_t between the normal friction and tangential friction is a primary factor to change locomotion shape of snake-like robot while all of the CPG parameters are constant.

Four groups of μ_n and μ_t such as (2.0, 0.4),(1.5, 0.3),(1.0, 0.2), and (0.5, 0.1) with equal ratio, are employed in simulation. By using identical CPG parameters in four cases,

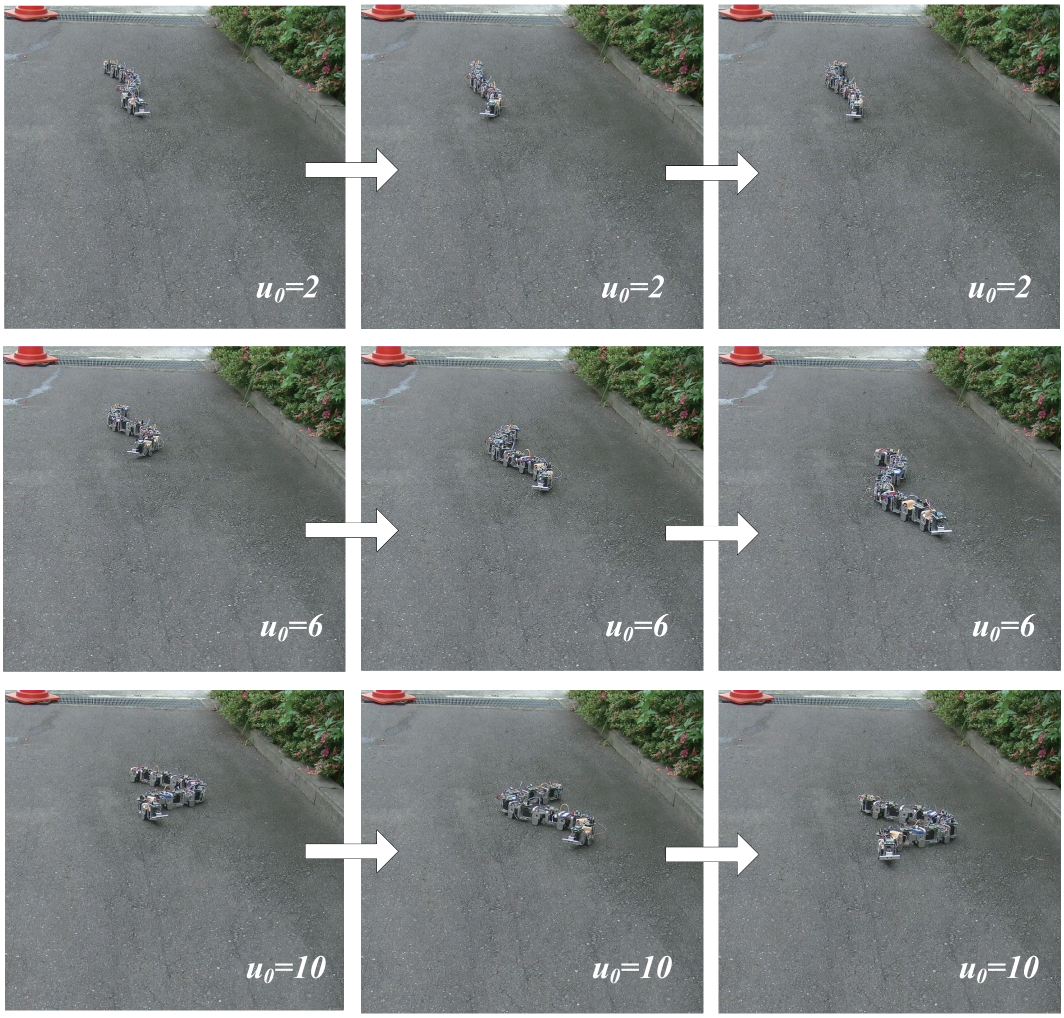


Figure 4.10. Scenes of the creeping motion on the slope with different driving input

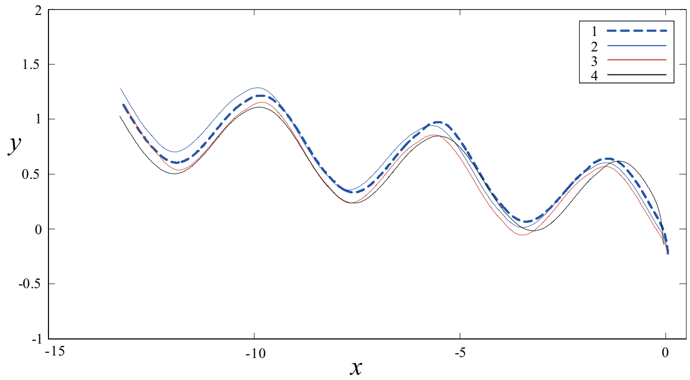


Figure 4.11. Trajectories of robot under four different friction

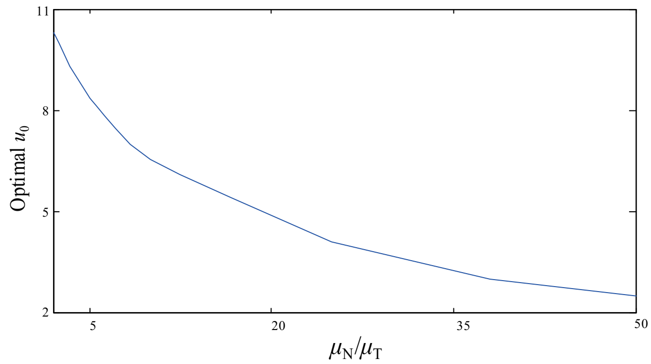


Figure 4.12. The relation between optimal u_0 and friction coefficient

the simulation results in Fig. 4.11 show that the trajectories of four different combinations of μ_n and μ_t are almost same. More groups of friction environment have been taken into simulation, the results of which have the same trend. The snake-like robot performs similar locomotion curvature when the coefficients of the normal friction and tangential friction have the same ratio, not depending on the coefficient value. Therefore, the value of optimal driving input u_0 which affects the locomotion curvature is mainly determined by the ratio of μ_n/μ_t. To get the highest locomotion efficiency, the driving input u_0 is optimized as shown in Fig. 4.12 when the ratio μ_n/μ_t varies from 2 to 50 with 25 sample points.

4.3 Summary

From the analysis of the characteristic of each CPG parameter, basic motion curvature, speed and numbers of S-shape can be controlled by adjusting driving input, time constant and different feedback CPG unit respectively. Furthermore, a curvature adaptive principle has been proposed for the motion optimization of the snake-like robot. To evaluate optimal creeping locomotion of the robot to changed friction or given slope, a criterion of locomotion efficiency has been introduced. Finally, based on the principle and criterion, the creeping locomotion of snake-like robot is optimized in the simulated experiment. Utilizing the relations between the optimal CPG parameters and different environment, the most adaptable locomotion can be achieved.

Chapter 5

CPG-based Self-adaptive Control

5.1 Adaptive Motion Control

In the Chapter 4, the basic motion patterns are all moving in symmetrical body waves. However, the body of the snake in nature performs various curves to move across uneven ground with interspersed large and small rocks or plants. For this reason, an asymmetric motion controller needs to be investigated for adaptive motion study [74].

To realize the asymmetric locomotion, we adopt the following control system. Each joint is treated as a correlative servo system, and the control system of the snake-like robot will therefore be constructed in a serial network structure. Initially, the snake-like robot creeps forward with symmetrical locomotion. When a locomotion modifying command from the high central nerve system is sent to the first joint, the motion configuration of the head joint would be altered. After a fixed interval, the modification of the first joint is shifted onto the second joint, and so on. The schematic of this kind of parameter transmission network is shown as Fig. 5.1.

Using this kind of motion control system, it is possible to transmit a wave all the way from the head to the tail, and realize an asymmetric locomotion of the snake-like robot without altering the whole creeping configuration. Furthermore, it allows smooth operation

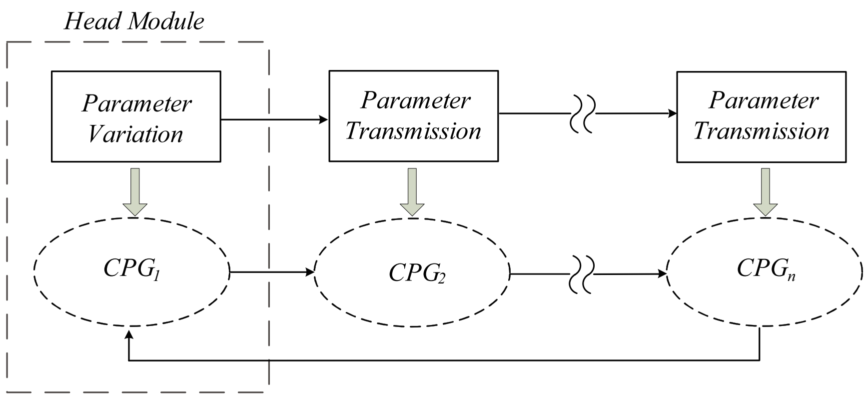


Figure 5.1. Schematic of the parameter transmission network.

of actuators and also closely matches the body curve of a real snake. This is a close approximation of the motor-neural mechanism of a snake [19].

Due to the fact that trajectory of each body segment which can follow the previous one in this locomotion control method, we need only plan the motion of the head module to obtain the desired locomotion. Based on this advantage, to couple the sensory information into the control system, we just need to collect the feedback signals from the head module.

Based on this asymmetric motion controller, three kinds of adaptive motion patters are introduced in this section.

5.1.1 Turn motion

A common behavior of most robots when they meet an obstacle in a forward direction is that they turn left or right to avoid the barrier. The snake-like robot can also perform a turn motion to avoid the obstacles. However, due to the characteristics of the creep motion of the snake-like robot generated by swinging the joints from side to side, the mechanism for performing a turn motion is totally different from that in wheeled drive robots.

When the rhythmic excitations exerted on each joint of the snake-like robot are sinusoidal waves, a symmetrical undulatory locomotion will be obtained. Since the winding angles to left and right balance out, the robot proceeds in a straight travel line on balance. However, as shown in Fig. 5.2, if the amplitude of wave in half period is altered from A to

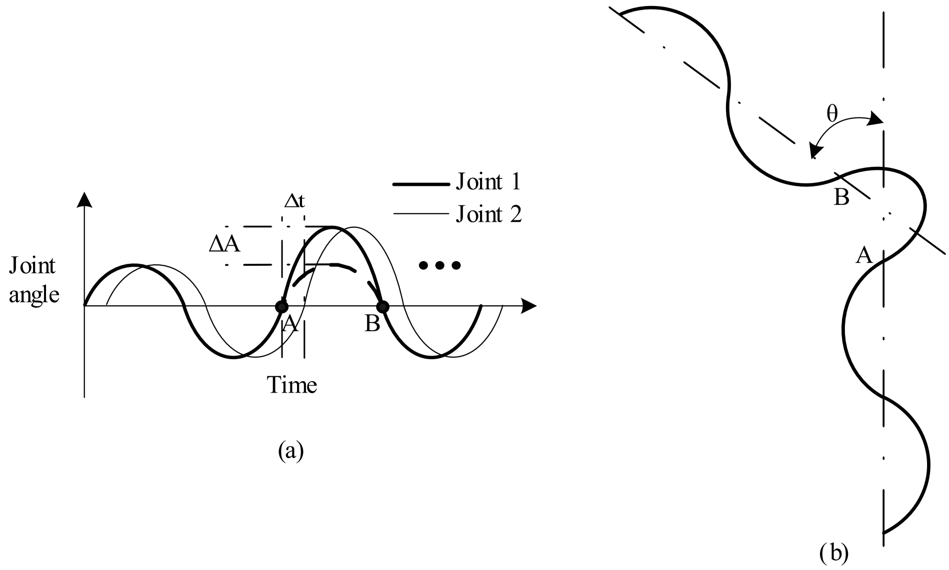


Figure 5.2. Turning motion with asymmetric locomotion (a) The angle signals of joints (b) The trajectory of the snake robot

B, this change will transmitted to the next joint successively after a interval Δt and thus the balance state will be shifted accordingly. Subsequently the overall moving direction of snake-like robot will be changed. Thus, a right or left turning motion can be executed just by exerting a positive or negative bias ΔA on the amplitude of the joints angle.

It is difficult to give a uniform rule like the left or right turn for wheeled drive robot to the snake-like robot. The turn direction of the creeping motion is mainly determined by two factors: the sign of bias ΔA and the direction of the head module at the time when signal begins shifting. If the orientation of the head is toward the right, a positive value of bias ΔA will carry out a left turn while a negative value of bias ΔA will cause a right turn. A completely reversed situation occurs when the orientation of head is toward the left. Herein the principle for controlling direction of the snake-like robot is shown in table 5.1

Table 5.1. The relation between the direction of robot and the controlled bias

The orientation of head	$\Delta A > 0$	$\Delta A < 0$
Right:	Turn Left	Turn Right
Left:	Turn Right	Turn Left

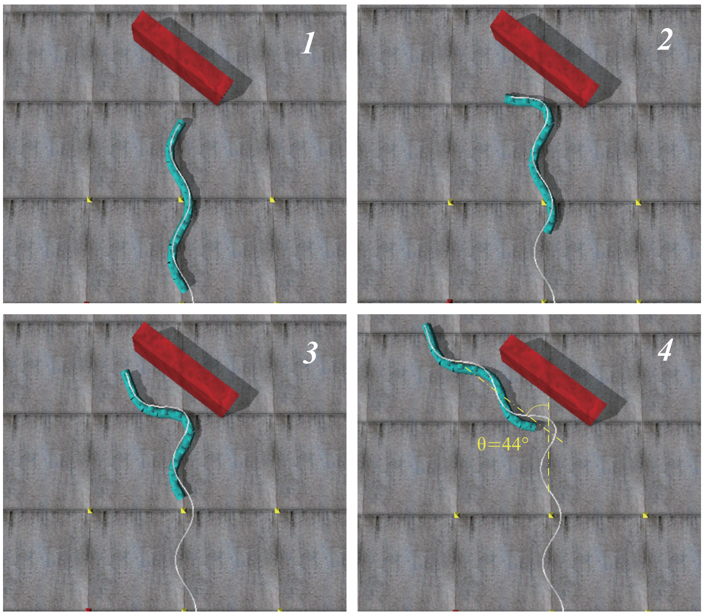


Figure 5.3. Simulation for turn motion

To testify the validity of the turn motion stated above, asymmetric locomotion of the snake-like robot for avoiding obstacle is conducted in the simulation, as shown in Fig. 5.3. By adjusting the parameter driving input u_0 of the head oscillator, the traveling direction of the snake-like robot is altered when the snake-like robot encounters an obstacle. The white line in the figure is the locomotion trajectory of the snake-like robot. Due to parameter-transmitting characteristic of the CPG network, the waveform of locomotion is changed from the first head segment to the tail segment one by one.

To obtain the desired direction of travel for the snake-like robot, we need to figure out the relation between the turn angle and the bias of the driving input Δu_0. A group of simulations are carried out with the same initial CPG parameters. How the Δu_0 affects the turning angle is shown in Fig. 5.4. As an example, Fig. 5.3 demonstrates a turn motion with the turning angle of 44° and driving input $\Delta u_0 = 3.5$.

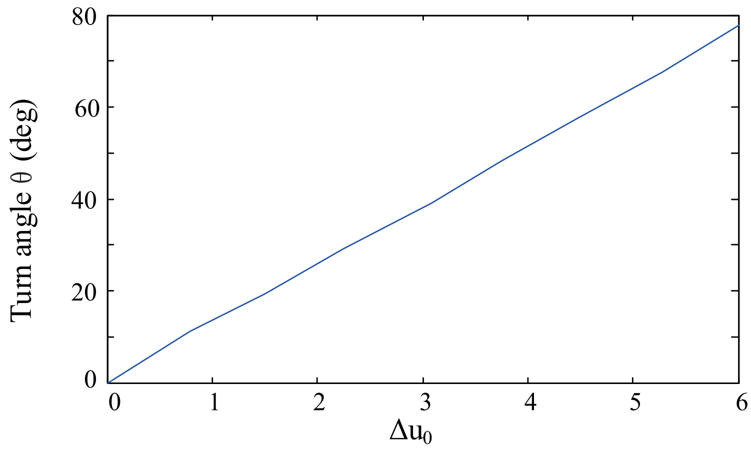


Figure 5.4. Turning angle with respect to different bias Δu_0

5.1.2 Round motion

With an elongated and limbless body, the natural snake can perform not only turn motion to avoid an obstacle, but also round motion to escape from irregular terrains. This is one of the most important characteristics of the natural snakes which are different from other animals. Utilizing this advantage of the snake, the snake-like robot can also creep through the barriers conveniently.

In order to achieve this kind of motion, the frequency of the excitation signal for the first joint angle is altered from A to B in half period, as shown in Fig. 5.5. This change is transmitted to the next joint consecutively in the same way as that in turning motion. From Fig. 5.5 (a), we can see that the increment of the signal period has brought a smaller phase difference between two adjacent joints angles. Thus the locomotion curve of the snake-like robot will have a larger diameter of the motion curve with smaller curvature. Consequently an asymmetric forward locomotion to pass around the obstacle is achieved.

Based on the above analysis of the CPG-based control for a round motion of the snake-like robot, a changed motion curve can be performed by adjusting the time constant τ_1 (τ_1/τ_2=const). Fig. 5.6 shows the scenes of a snake-like robot rounding a cylindrical obstacle. The gliding curve of the robot body also changes from the head segment to the tail

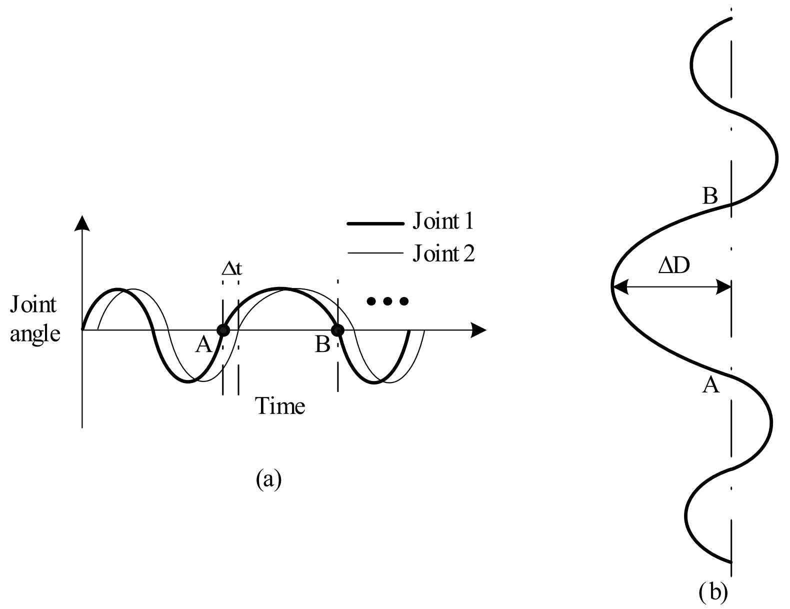


Figure 5.5. Round motion with asymmetric locomotion (a) The angle signals of joints (b) The trajectory of the snake robot

consecutively. From the white line in the figure, it is known that the motion of snake-like robot with an asymmetric curve has exhibited a smooth trajectory.

The time constant $\Delta\tau$ (τ_1/τ_2=1/3) keeps a particular relation with the maximum distance ΔD between the motion curve and the forward direction, as shown in Fig. 5.7. Thus, the snake-like robot can creep with an adaptive gliding curve to avoid obstacles by adjusting the time constant suitably. The demonstration in Fig. 5.6 is a round motion with $\Delta\tau = 2.0$ and $\Delta D = 0.21m$.

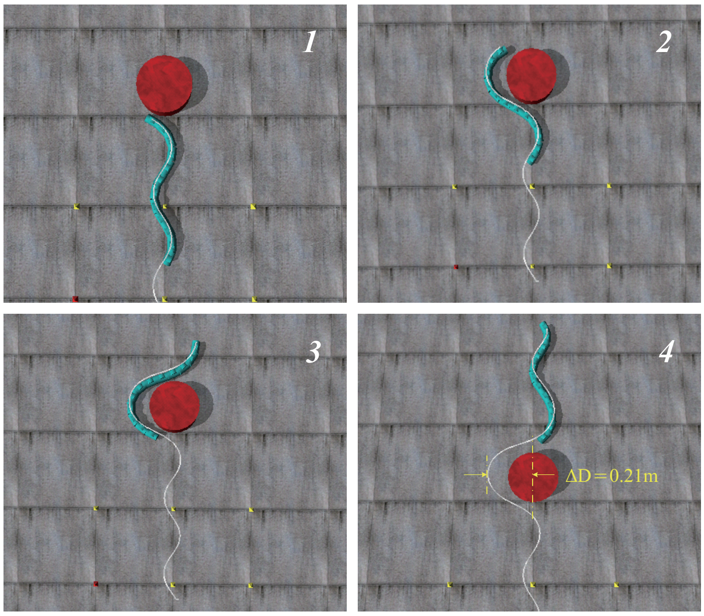


Figure 5.6. Simulation for round motion

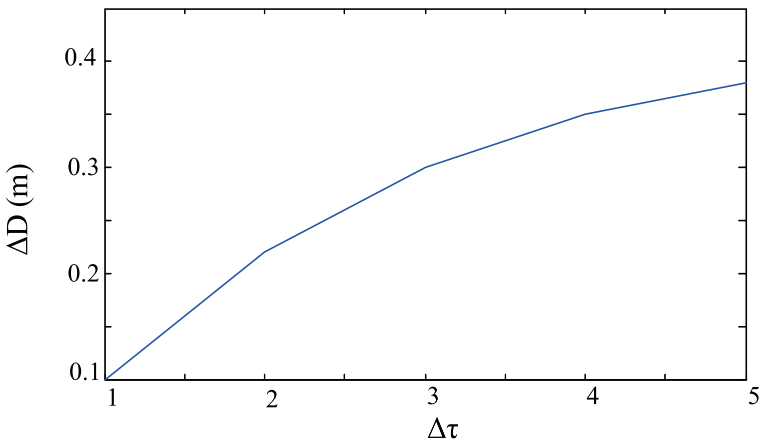


Figure 5.7. Maximum distance of the motion curve with respect to different bias $\Delta\tau$

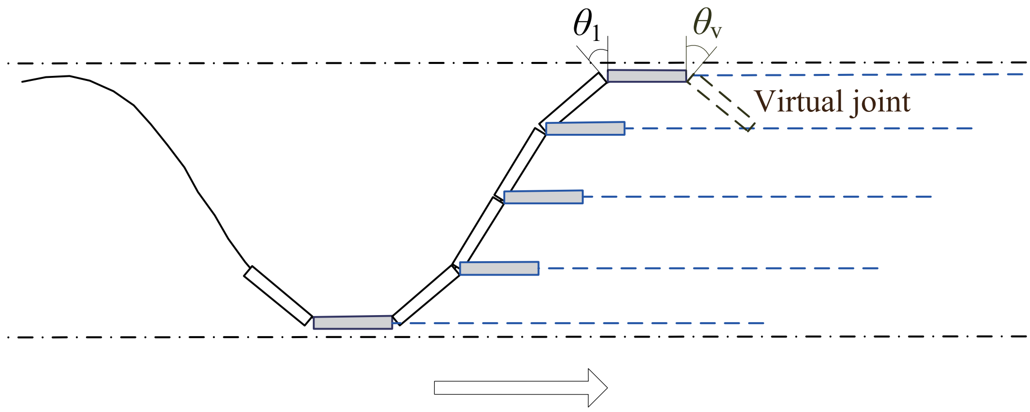


Figure 5.8. Head-navigated locomotion of the snake-like robot

5.1.3 Head-navigated locomotion

Inspired from the natural snake, the head of the snake-like robot would be controlled to lift up and keep the same orientation as the forward direction. A control scheme is needed to make the robot maintain the direction of its head during undulating locomotion. Yamada in [83] has studied the stabilization of the head of an undulating snake-like robot. A sin-based continuous model and discrete model have been analyzed elaborately for the motion control. Herein, due to the different CPG-based control model, we will give another approach to realize the head control of the snake-like robot [77].

The concept of the head-navigated locomotion is shown in Fig. 5.8. The blue squares represent the head of the snake-like robot during the motion. Despite the S-shape motion pattern, the head module can always point to the direction of the forward movement. The dash line means the detection area of a distance sensor which is installed on the head. It can be found that the sensory information in this kind of locomotion will just explore the front area which the robot will pass through.

In the traditional serpentine locomotion, the angle of head joint also swings from side to side just as the other joints on the robot body. To obtain a head-navigated locomotion, the angle of the first joint which connects the head and second link should be controlled by a totally different model. There are three basic features of the angle of the first joint:

(1) The direction of the head module should always be kept the same as the forward direction of robot.

(2) The angle of the first joint is also changed periodically and has an identical frequency as the body joints.

(3) The angle of the first joint has the same kind of periodic signal as the body joints but different phase and amplitude.

The turning point from the original head control to the head-navigated control is the problem that we are concerned with. As shown in the Fig. 5.8, we add a virtual joint before the head, and set the angle of it having the same phase difference with the first joint as that between the first joint and the second one. By using this virtual joint as a reference, when the head module is located at the situation where $\theta_1 = \theta_v$, the direction of head will just be facing the forward direction of the snake-like robot. Furthermore, by analyzing the relation between the original head joint angle and the desired head joint angle, four typical situations of the joints during the head-navigated locomotion have been shown in Fig. 5.9.

In the Fig. 5.9, θ_1 is the original angle of the first joint under normal S-shape locomotion; θ_2 is the angle of the second joint; θ_h is the modified angle of the first joint under head-navigated locomotion. In *situation (a)*, where the original value of first joint θ_1 is the same as second joint θ_2, θ_h should be zero to let the head link point to the forward direction. Meanwhile, $\theta_1 - \theta_2$ is also zero at this time. In *situation (b)*, where the original value of first joint θ_1 just equals the inverse value of second joint θ_2, θ_h comes to its maximal value. For the value of $\theta_1 - \theta_2$ also get to the Maximum. Due to the feature (2), the configuration of the joints in situation (c) and (d) are similar with that of situation (a) and (b) which are just different in phase. These four typical situations compose a whole period of locomotion. The relations of each angle signals are clearly shown in Fig. 5.9 (e). From these results, it is obvious to find that the angle signal of θ_h has the same phase and frequency as those of $\theta_1 - \theta_2$. Since the same kind of periodic signals as stated in feature (3), the angle signal of θ_h can be expressed as a function of $\theta_1 - \theta_2$. If we define the original function of the angle of the first joint as $f(t)$, and phase difference of two adjacent joints as φ, the equation of

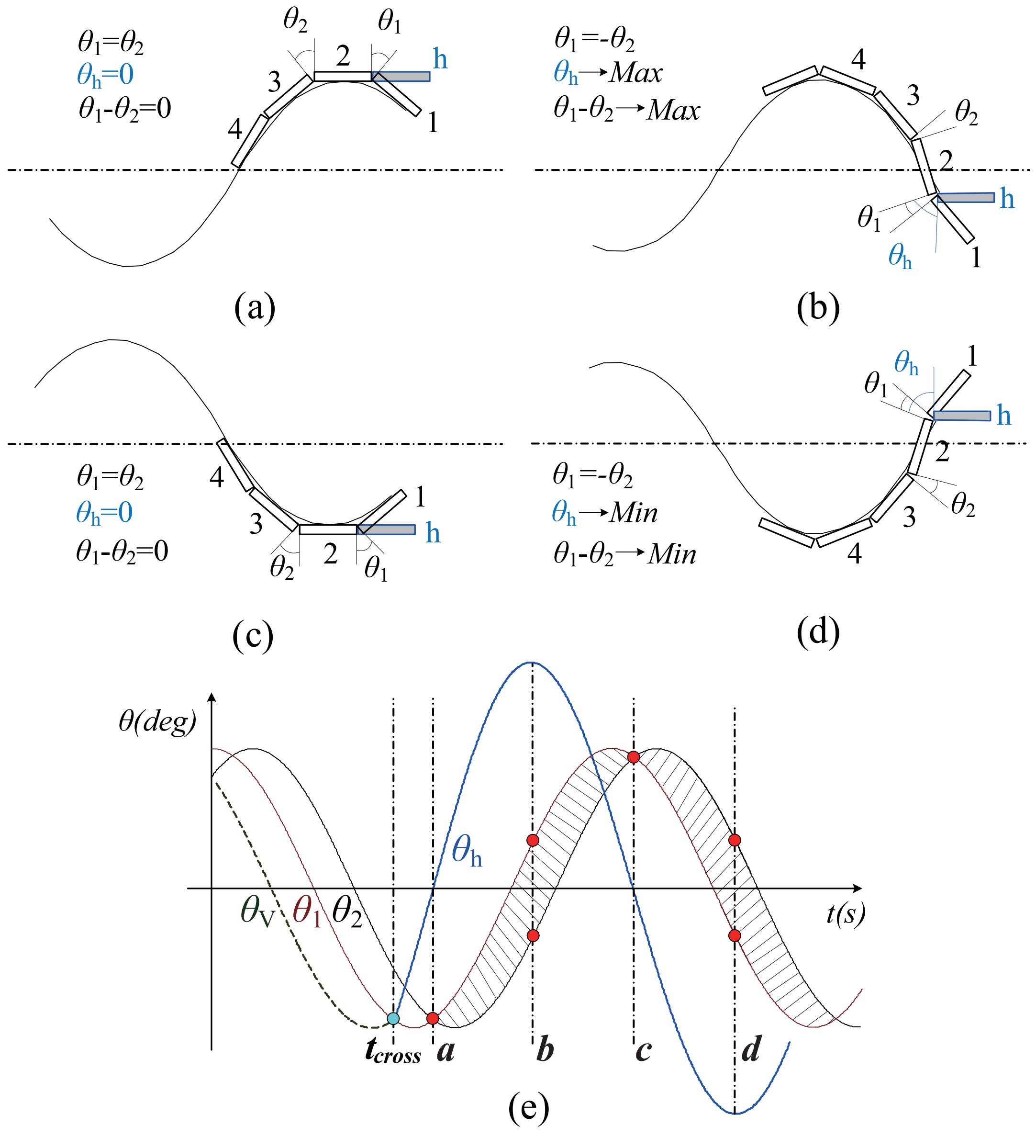


Figure 5.9. Configurations of the joint angles in the head-navigated locomotion of the snake-like robot

desired head joint angle θ_h could be described as follows:

$$\begin{aligned}
&\theta_1 = f(t) \\
&\theta_v = f(t+\varphi) \\
&\theta_2 = f(t-\varphi) \\
&\theta_h = A(\theta_1 - \theta_2) = A[f(t) - f(t-\varphi)]
\end{aligned} \tag{5.1}$$

From the above equations, the major problem in control of the head joint has changed to solve the value of amplitude coefficient A. From Fig. 5.9 (e), the crossing time t_{cross} of θ_1 and θ_v can be calculated easily. Meanwhile, due to θ_1 equals θ_h at this time, the value of amplitude coefficient A can be obtained by:

$$A = f(t_{cross})/[f(t_{cross}) - f(t_{cross} - \varphi)] \tag{5.2}$$

Based on the analysis of the head-navigated locomotion in section III, the configuration of the first joint in the snake-like robot needs to be redesigned separately. A virtual CPG is needed to be added for the judgment of the start point of head-navigated motion. From the equation (5.1) and equation (5.2), the control signal of the head joint $y_{out,head}$ is calculated from the output of the first CPG $y_{out,1}$ and the second CPG $y_{out,2}$ given in equation (5.3). A is an amplitude coefficient which can be calculated from the crossing point of $y_{out,1}$, $y_{out,virtual}$ and $y_{out,head}$ at time t_{cross}. One simple demonstration of the control signals from the CPG network is shown in Fig. 5.10.

$$\begin{cases}
y_{out,head}(t) = A[y_{out,1}(t) - y_{out,2}(t)] \\
y_{out,1}(t_{cross}) = y_{out,virtual}(t_{cross}) \\
y_{out,1}(t_{cross}) = y_{out,head}(t_{cross}) \\
A = y_{out,1}(t_{cross})/[y_{out,1}(t_{cross}) - y_{out,2}(t_{cross})]
\end{cases} \tag{5.3}$$

In order to verify the proposed control method for the head-navigated locomotion, a simulation has been performed in simulation environment. The rhythmic output of the CPGs are implemented on the joints of the robot as the angle input signals. The simulation results are shown in the Fig. 5.11. The locomotion in the 1st and 2nd scenes is still a

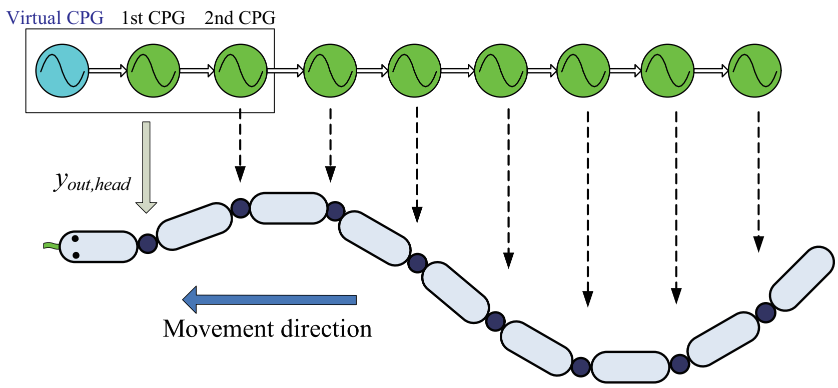


Figure 5.10. CPG network implemented to control a head-navigated locomotion

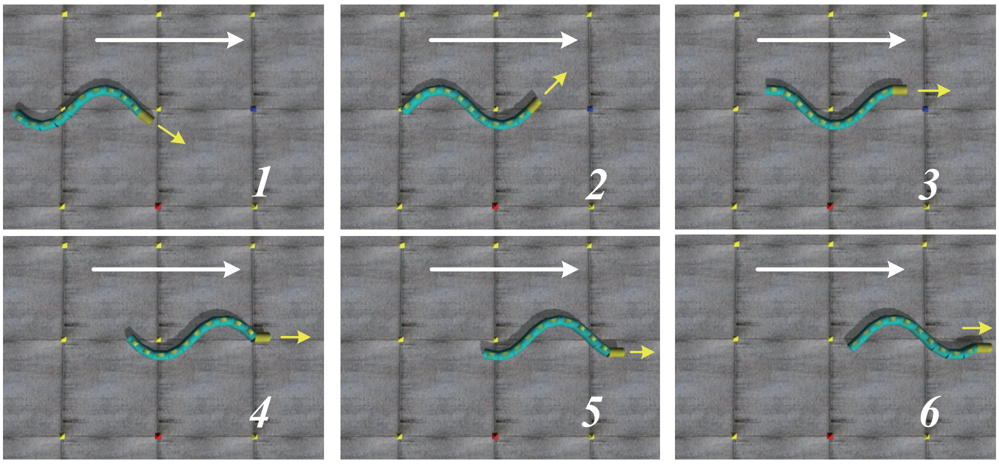


Figure 5.11. Simulation results of the head-navigated locomotion

traditional serpentine movement where the head module is swinging from the side to side. The 3rd scene is the turning point where head module just has the same direction as that of forward movement. From this point, the motion will change into head-navigated locomotion smoothly, and maintain its orientation as shown in 4th and 5th scenes. The control signals of joint angles are shown in Fig. 5.12. The dash-and-dot line between the red line and the blue line represents the turning point from the original head motion to the head-navigated motion, which corresponds to the 3rd scene of Fig. 5.11.

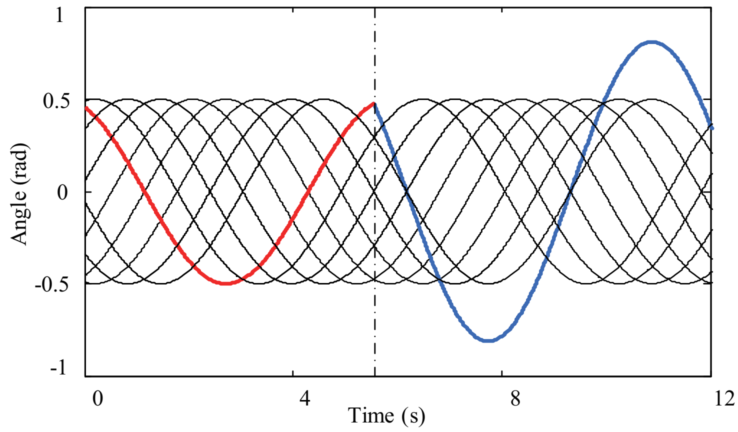


Figure 5.12. Control signals of the joints in the simulation. Red line and blue line represent the angle signal of the first joint in two different motion patterns. Black lines are the angle signals of other joints.

5.1.4 Experiments

To testify the validity of CPG-controlled obstacle avoidance, experiments were carried out on our snake-like robot *SR*-I.

Turn and Round Motion

By implementing rhythmic outputs derived from the CPG-based control system onto the driving joints, the snake-like robot successfully exhibits creeping locomotion. As shown in Fig. 5.13, an asymmetric locomotion curve of the snake-like robot is realized to avoid obstacles with turn motion. A round motion of the snake-like robot is also conducted experimentally as shown in Fig. 5.14. In the experiments, a continuous locomotion curve of the snake-like robot is performed during the transmission of angle signals from the head joint to the tail joint.

The snake-like robot has been proved to be successfully in performing adaptive locomotion by using the proposed CPG-based control method. However, the adaptive locomotion control that we present here is discussed separately. In order to realize reactions to more complex situations, an asymmetric locomotion of the snake-like robot coupling with both turn motion and round motion synchronously should be investigated in our future studies.

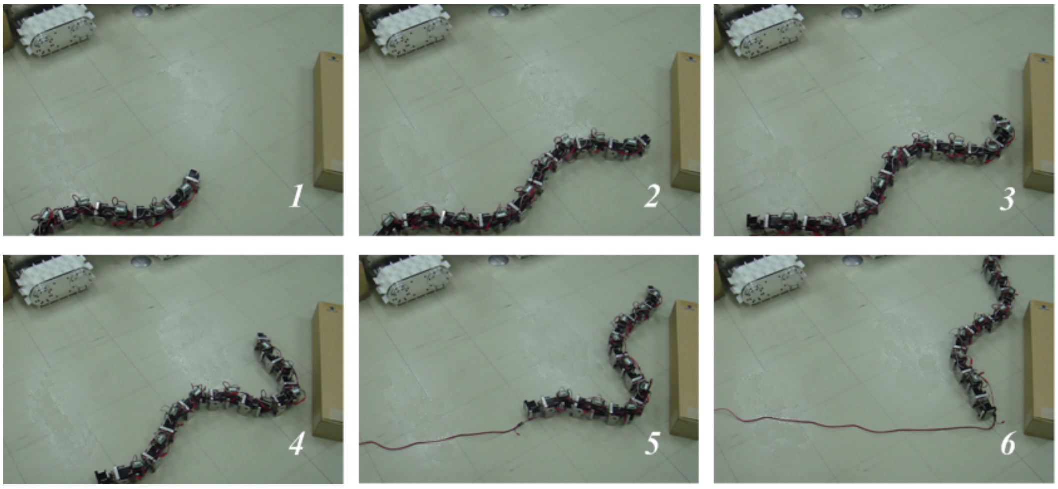


Figure 5.13. Experiment scenes of the snake-like robot to avoid obstacle with turn motion

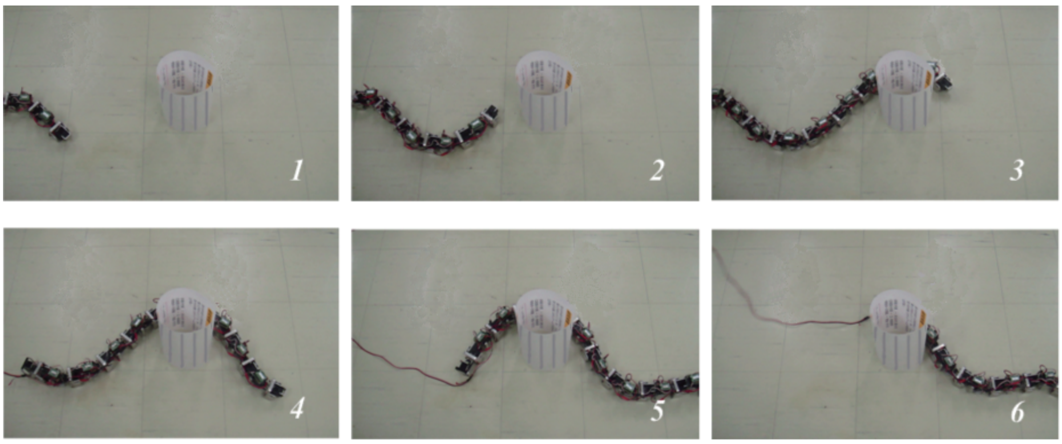


Figure 5.14. Experiment scenes of the snake-like robot to avoid obstacle with round motion

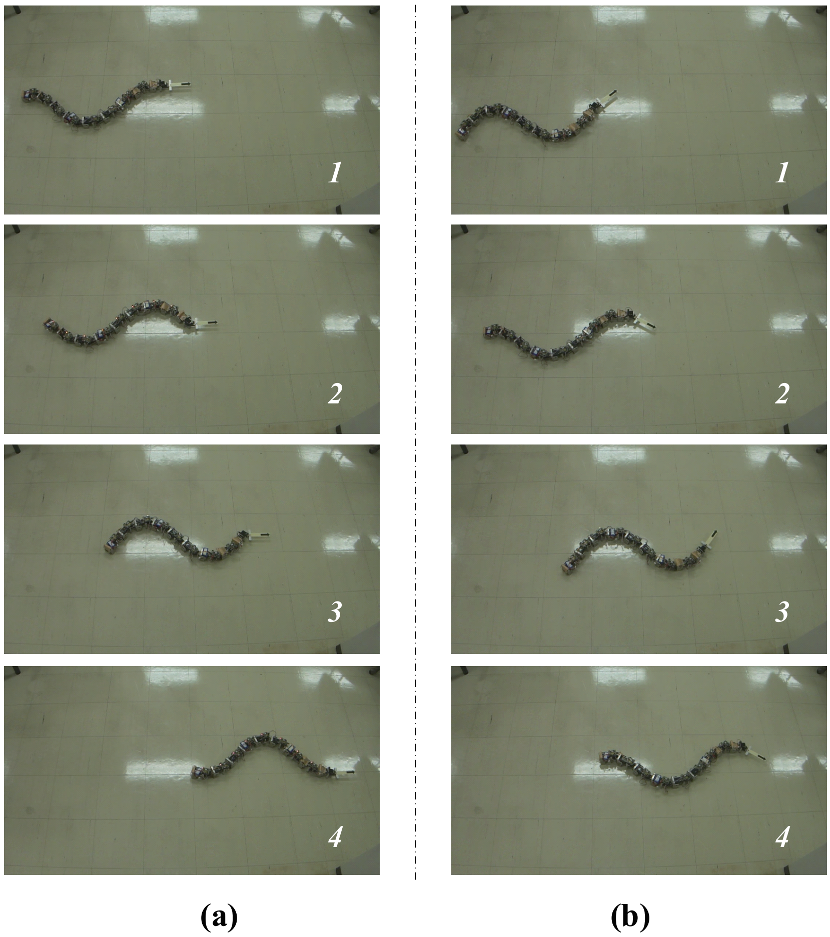


Figure 5.15. (a) Experiment scenes of the head-navigated locomotion; (b) Experiment scenes of the normal serpentine locomotion

Head-navigated Motion

By implementing normal control method and modified head-navigated method onto the control system, the snake-like robot exhibits different motion pattern as shown in Fig. 5.15. From the scenes of experiment, we can find that the head can always have the same direction as the forward direction during the head-navigated locomotion, while the normal serpentine locomotion do not have this property.

5.2 Modeling of AMM-based Turn Motion

Due to the characteristic of the creep motion is generated by swinging the joints from side to side, the mechanism to perform a turn motion is totally different from that of wheeled robots. To obtain the relation between the parameters of the joint angle and the turn motion, here we will give an analysis of the AMM-based turn motion of the snake-like robot.

5.2.1 Amplitude Modulation Method

The Amplitude Modulation Method has been widely used to realize the turn motion of the snake-like robot [76]. It means the amplitude of the joint angle will be added a bias value in half period, and an asymmetric swing of joint will lead to a change in the motion direction. As shown in Fig. 5.16, when the symmetrical signals are exerted on each joint of the snake-like robot, the robot proceeds in a straight travel line. If the amplitude of head angle is altered from point A, the balance is broken and the motion direction is changed. This change will transmitted to the next joint successively after a interval Δt. While coming to point B, the amplitude of the angle will return to the previous value as that before point A. And then the robot will move along a new direction. The whole progress mentioned above is the turn motion in one period. Here it should be mentioned that we only consider to add a bias when the signals is at zero point, so that there is no any sudden or discontinuous change of joint angle.

From the turning trajectory shown as Fig. 5.16 (b), we can find that the winding angle α of the body curve can be used to express the turning angle of the snake-like motion. Three different dash lines show the axis direction under different amplitude of joint angle. During the turn motion, the turning angle of the snake-like robot can just be calculated from the angle between the axis 1 and 3. Due to the same configuration of robot moving in the axis 1 and 3, the winding angles in these two cases are equal to α_0. For axis 2, we

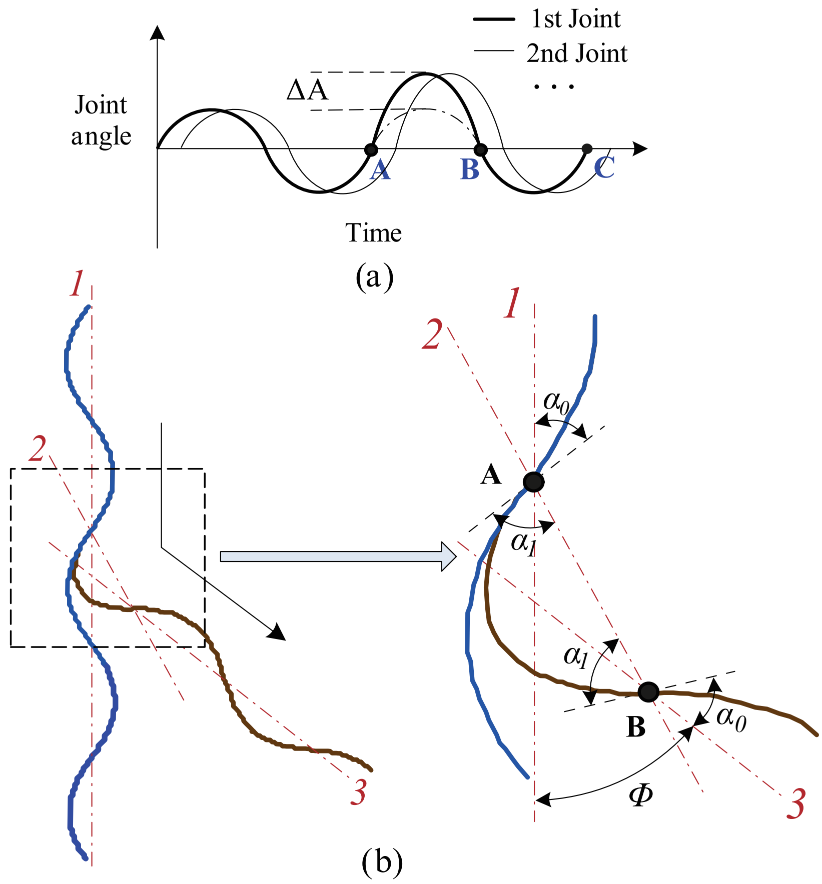


Figure 5.16. Turn motion of the snake-like robot. (a) The angle signals of joints (b) The trajectory of the head link during the locomotion

define its winding angle is α_1. Thus, the exact turn angle Φ can be expressed as follows:

$$\Phi = 2(\alpha_1 - \alpha_0) \tag{5.4}$$

5.2.2 Computational Model

Hirose [19] has already studied snakes and found that their bodies take on the so-called serpenoid curve when they locomote with a serpentine gait. Fig. 5.17 shows the gliding curve of the snake creeping locomotion. The serpenoid curve is a curve whose curvature changes sinusoidally along the axis of the curve, and sinusoidal configuration of the joint angle approximates the serpentine motion. s is a distance along axis of the body and $\theta(s)$ is the distribution of the bending angle along body axis. The bending angle at position s

with amplitude A is given by

$$\theta(s) = A\sin(\frac{\pi}{2} \cdot \frac{s}{l}) \tag{5.5}$$

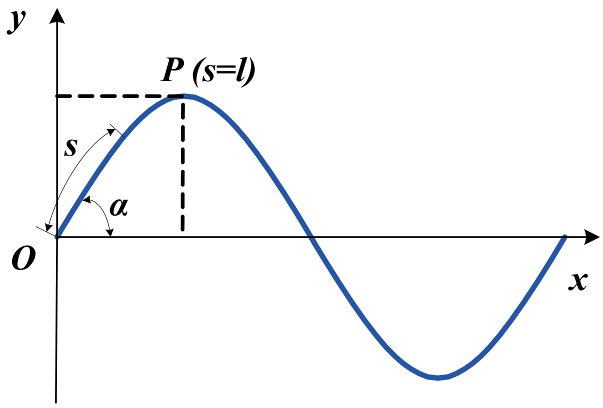


Figure 5.17. Gliding curve of the snake creeping locomotion

Here, l is the length of the quarter-cycle of the curve. α is defined as winding angle which is the integration of the bending angle from 0 to P. If we make the curve consist of line elements of incremental length δs (corresponding to snake vertebrae), α can be expressed as

$$\alpha = \frac{1}{\delta s}\int_0^l \theta(s)ds \tag{5.6}$$

If we substitute equation (5.5) into equation (5.6), the wingding angle α can be calculated by

$$\alpha = A\frac{l}{\delta s}\frac{2}{\pi} \tag{5.7}$$

From the above analysis, we know that the turning angle of the robot can be expressed as a function of winding angles. If we define the initial amplitude of the joint angle is A, and the bias of the amplitude for turn motion is ΔA, by substituting equation (5.7) into equation (5.4), it can obtain

$$\begin{aligned}\Phi &= 2(\alpha_1 - \alpha_0) \\ &= 2((A + \Delta A)\frac{l}{\delta s}\frac{2}{\pi} - A\frac{l}{\delta s}\frac{2}{\pi}) \\ &= \Delta A\frac{l}{\delta s}\frac{4}{\pi}\end{aligned} \tag{5.8}$$

From this equation, it can be found that the turn angle is linearly influenced by the bias of the amplitude when the body parameters is constant. This is very important for the control of the turn motion. It means that we can easily steer the motion direction of the snake robot by adjusting the amplitude of joint angle.

To get a more general computational model of the AMM-based turn motion, the different numbers of the S-shape will also be taken into consideration. While l is the length of the quarter-cycle of the curve, it can be represented as

$$l = \frac{n\delta s}{4N} \tag{5.9}$$

where N is the numbers of S-shape, n is the numbers of links, and δs is the length of each link. By substituting into equation (5.8), a more general model can be obtained as

$$\Phi = \frac{n}{N\pi}\Delta A \tag{5.10}$$

For a desired turning behavior, the change of the angle amplitude in the snake robot can be calculated from equation (5.10) conveniently. Furthermore, we know that the amplitude of joint angle is influenced linearly by the driving input u_0 of the CPG network. If η is the coefficient of the linear relation between the joint amplitude A and the CPG driving input u_0, the desired turn motion can be achieved by equation (5.11). Therefore, AMM-based turn motion can be easily conducted in our designed CPG-based oscillator network.

$$\Phi = \eta\Delta u_0 \frac{n}{N\pi} \tag{5.11}$$

5.2.3 Features of AMM

Based on the equation (5.10) which reveals the relation between the motion parameters and the turn angle, some important features of turn motion by AMM are concluded. Firstly, if the basic configuration of the snake-like robot is unchanged, the magnitude of the turn angle is decided by the bias of the amplitude of the joint angle linearly. Fig. 5.18 (a) shows a group of the trajectories of the turn motion with different bias of the amplitude. It can be found that the turn angle is increased linearly with respect to the change of the amplitude

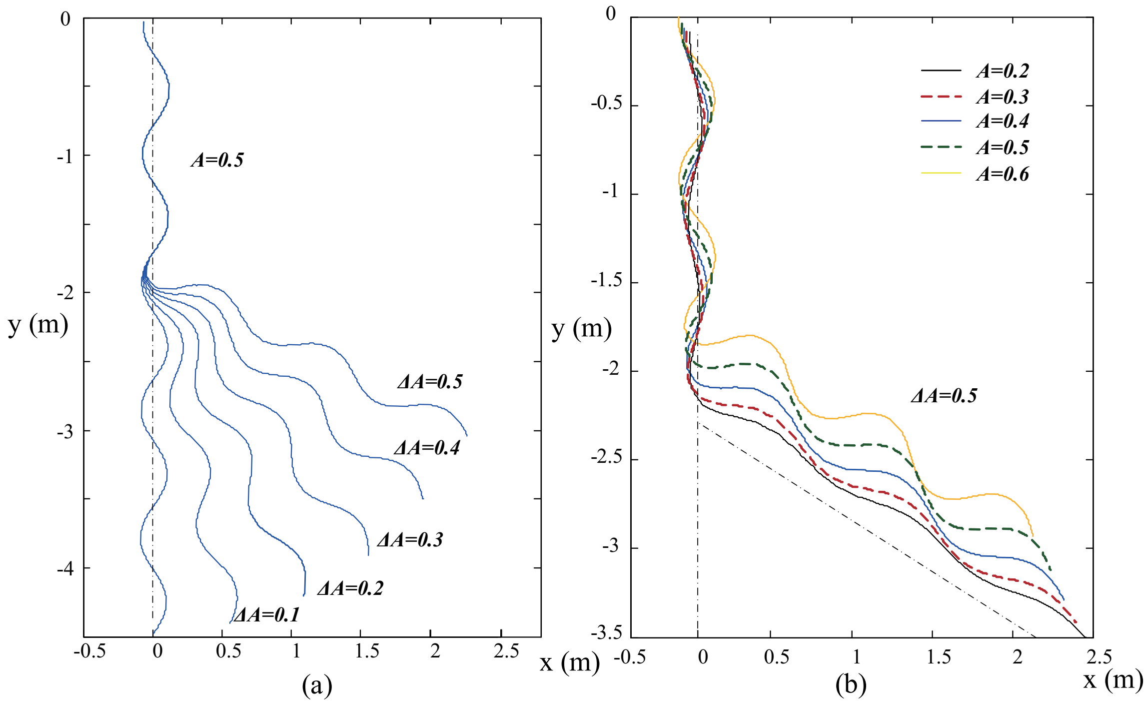


Figure 5.18. Features of the AMM-based turn motion

ΔA. Secondly, no matter how different the initial value of the amplitude of the joint angle is, the turn angle is just influenced by the offset of it. This characteristic can also be verified from Fig. 5.18 (b), where the trajectories of the turn motion are generated with different initial amplitude but the same bias. We can see their turn angles are almost the same.

To obtain a smooth and continuous change of the joint angles, the adjustment of the amplitude of the joint angle is implemented while the angle is at zero value point. Thus, the turn motion can not start from any time of the creeping motion, but just start at the time when angles come to zero. There are two kinds of cases for the zero point: one is changed from positive angle to negative angle; another is changed from negative angle to positive angle. The increase or decrease of the amplitude of joint angle in these two cases have an opposite influence on the turn direction. If we define a counterclockwise joint angle is positive and a clockwise one is negative, two different cases for the zero point can be expressed by the sign of derivative of angle. Equations for left turning or right turning can be obtained as equation (8). This equation has explained that the bias of the amplitude

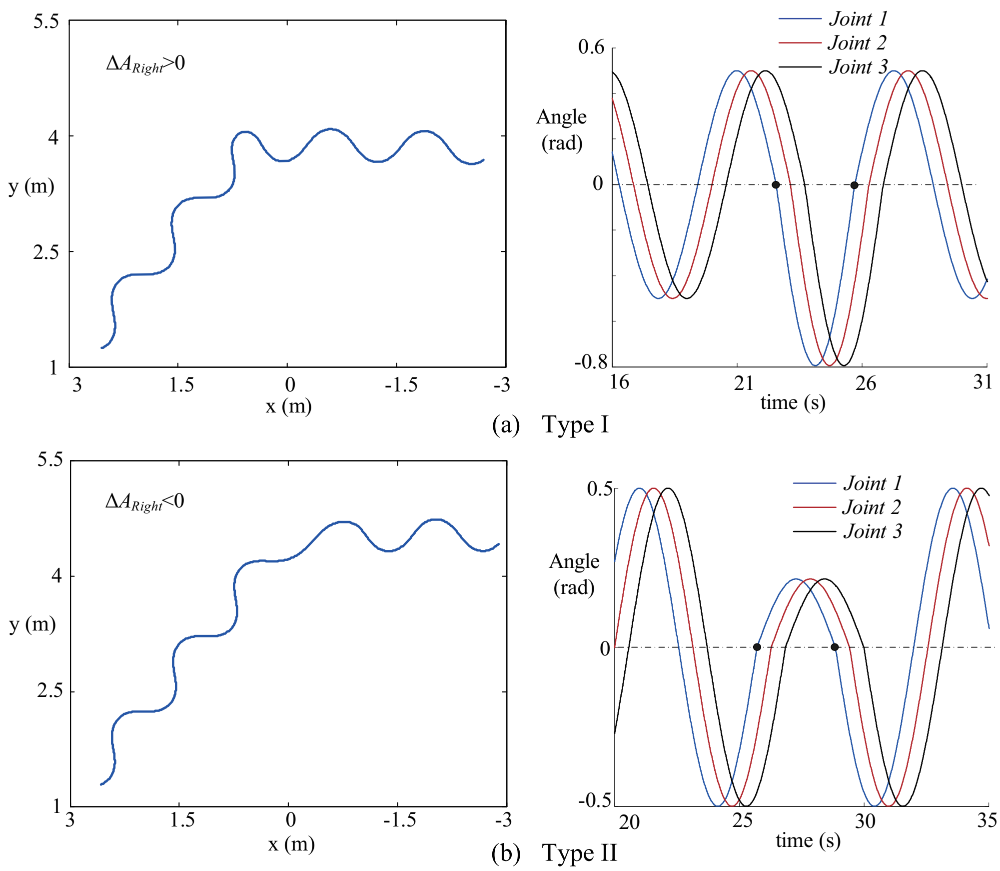


Figure 5.19. An example of right turning motion in two different cases

should be positive or negative with respect to the turn direction.

$$\begin{cases} \Delta A_{Right} = -sign(\dot{\theta})\Delta A \\ \Delta A_{Left} = sign(\dot{\theta})\Delta A \end{cases} \tag{5.12}$$

Fig. 5.25 has shown an example of right turning motion in two different zero points. The left part of the figure is the trajectory of the turn motion, and the right part is the waves of the angle signals. From the angle waves of the joints, the increase and decrease of the amplitude has led the same right turning which verified the equation (8). For easily explaining, the turn motion caused by a increased amplitude is called "type I", and the turn motion caused by a decreased amplitude is called "type II".

It should be mentioned that the above discussion of the turn motion of the snake-like robot is just considered for turning in one period. If the amplitude modification is applied on

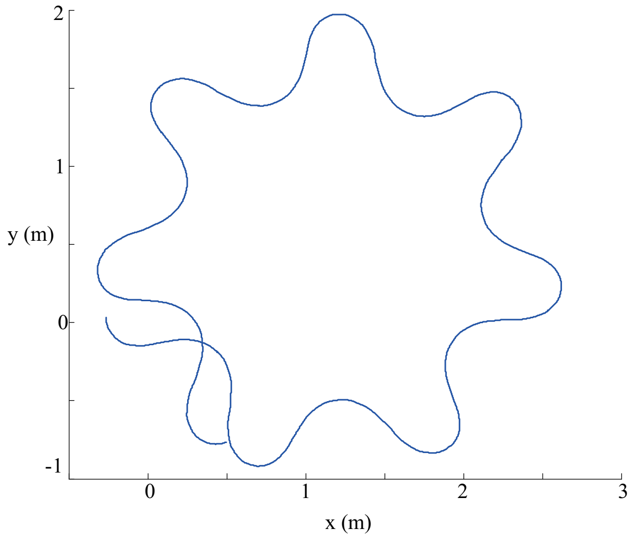


Figure 5.20. Circular motion trajectory of the turn motion

the robot joints continuously, the snake-like robot will keep turning and generate a circular motion trajectory as shown in Fig. 5.20. This kind of phenomenon has also been studied in some other researches [84][24].

5.3 Sensor-driven Neural Controller

5.3.1 Design of Collision-free Behaviors

The possibility of reactive behaviors of a bio-robot reveals the ability to perceive and act within the environment for a meaningful and purposive manner [1]. A study of intelligent behaviors of a snake-like robot is necessary during the design of the motion control system. Based on the sensory information, the work of the recognition of unstructured environment and the selection of adaptive motion can be achieved through the reactive behaviors of the robot [79].

Collision-free behavior is one of the most important reactive behaviors of the bio-robot. From the above study of the CPG-based motion control of the snake-like robot, the basic motion patterns have been analyzed. To get adaptive locomotion, the collision-free mo-

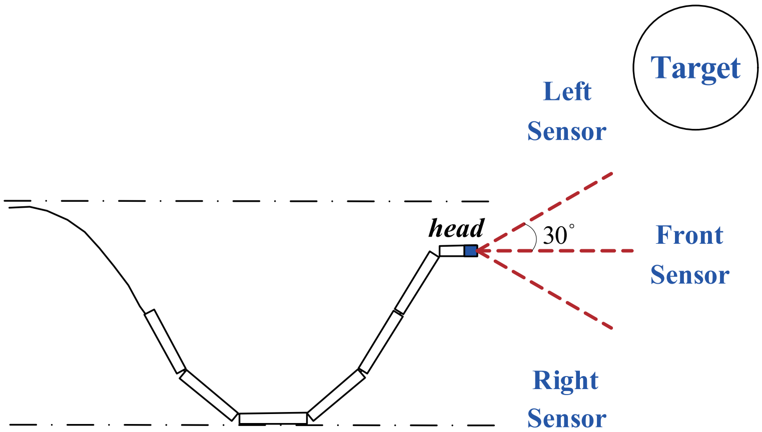


Figure 5.21. Sensor arrangement with head-navigated locomotion

tion pattern of the robot corresponding to certain circumstance should be investigated. The acquirement of the environmental information during the movement is depended on the sensory system. With the help of the proposed head-navigated locomotion, the environmental condition in front of the undulatory body could be detected conveniently by distance sensors installed on the head module. Fig. 5.21 shows the arrangement of the sensors for obstacle detection. The left, right, and front sensors can be used to detect the distance between the head and obstacle. Furthermore, the target command is also considered for the motion planning of the robot. Herein the target information will work as a high level command, while the sensory signals will conduct the low level effectors.

To design a self-adaptive controller for the collision-free behavior, the relations between the sensory information and the reactive motion patterns need to be figured out. All possible situations corresponding to the different kinds of motion strategies should be summarized. Based on the summarized results of the robot behavior, the topology structure of the sensor-driven neural network can be constructed. To provide a clear explanation, we use the schematic diagram shown in Fig. 5.22 to explain the motion strategies with respect to different situations.

To illustrate the scheme conveniently, the target information is divided into two situations: Left Target (LT) or Right Target (RT) meaning the robot will move to the left or right. Coupled with three sensors: Left Sensor (LS), Front Sensor (FS) and Right Sensor

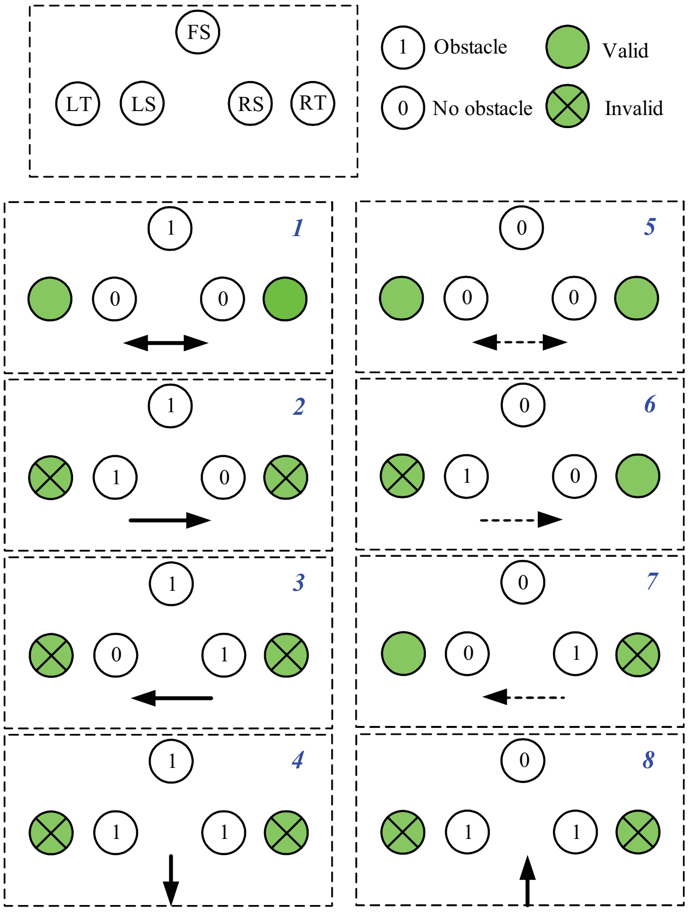


Figure 5.22. Scheme for the motion strategies in different situation. LS: Left Sensor, FS: Front Sensor, RS: Right Sensor, LT: Left Target, RT: Right Target

(RS), there are a total of five inputs to decide the motion of the snake-like robot. As shown in Fig. 5.22, we use 1 and 0 to indicate whether the IR range sensor detects an obstacle. A circle with a cross indicates the target command is blocked, while a circle without the cross means the target command is valid. In other words, if an obstacle is just blocking the target direction, the snake-like robot will perform collision-free behavior preferentially and ignore the target command. The arrow shows the desired moving direction corresponding to different situations. The arrow with a solid line means the robot must turn in that direction. The arrow with a dashed line means the turning motion is not necessary, but it depends on the target signal.

Here we give two examples to describe these illustrations. In the sixth illustration of Fig. 5.22, if there is not a target command the robot will just go forward. But only the target signal in the right side can be valid since the left side has obstacles. In the fourth illustration the robot have to perform escape behavior and go backward due to the obstacles has blocked all the ways. Meanwhile, the target command in both sides are invalid. From the analysis result, it can be found that there are eight groups of situations for the snake-like robot to perform collision-free behaviors. The neural preprocessing network will be designed according to these sensor-driven reactive behaviors.

5.3.2 Design of Neural Controller

The design of the sensor-driven neural controller will be introduced in this part. Based on the above discussion, bio-inspired Neural Oscillator Network (NON) has been employed in the control of the snake robot. To achieve an autonomous collision-free behavior, local information from the environment will be considered to have a closed-loop control system. CPG receives inputs from a higher level of the central neuro, and also from peripheral receptors. Thus, its functioning results from an interaction between central commands and local reflexes [18]. This concept allows us to integrate the sensory signals into the CPG oscillator network to drive the numbers of joints for self-adaptive locomotion.

A brief flow diagram of the reactive motion controller for the snake-like robot is shown in Fig. 5.23. In particular, situation where the adjustment of motion with a local sensor-based approach may be prevented by a global human-based approach is a significant challenge in the design of the controller. A network called Neural Preprocessing Network (NPN) is proposed as the approach to solve the problem of signal fusion. The control system of the snake robot can integrate the low-level sensory information and high-level target commands, and evolves to an automatically optimal configuration on its own. Signals from peripheral receptors that are related to the reactive motion can be set as the input of the neural network after scaling. The output of the NPN will be adopted to adjust the driving input of the NON and achieve the desired motion pattern. For example, a modification of the driving input can achieve a AMM-based turn motion to perform a collision-free behavior.

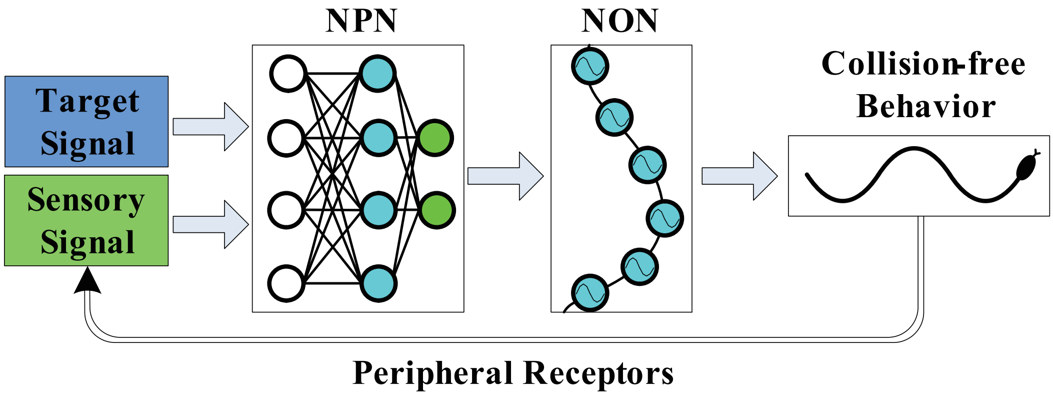


Figure 5.23. Flow diagram of the sensor-driven neural controller

5.3.3 Neural Preprocessing Network

The design of the neural network is based on the analytic results in sensor-driven reactive behavior of the snake robot. The topology structure of the neural network is divide into three layers. The sensory signals are input layer. In total, there would be five signals needed to feed back into the neural preprocessing network. The middle layer working as signal processing network. To simplified the structure, the neurons in the middle layer with only one connection from the input layer are eliminated and substituted by direct connections with the output neurons. Finally, the output layer generates the results of the NPN to adjust the driving input of the CPG motion controller. The topology of the network is shown in Fig. 5.24.

The neuron model in the NPN also comes from Matsuoka's neuron model [40]. The single neuron can be expressed by the following equations:

$$
\begin{aligned}
&\tau_1 \dot{u} + u = T - \beta v - \sum S_{input} \\
&\tau_2 \dot{v} + v = y \qquad (5.13) \\
&y = g(u) = max(0, u)
\end{aligned}
$$

where u is the membrane potentials of the neuron; v is the variable that represents the degree of adaptation; y is the output of the neuron; T is the threshold value of the neuron; τ_1 and τ_2 are the parameters that specify the time constants for the membrane potential and

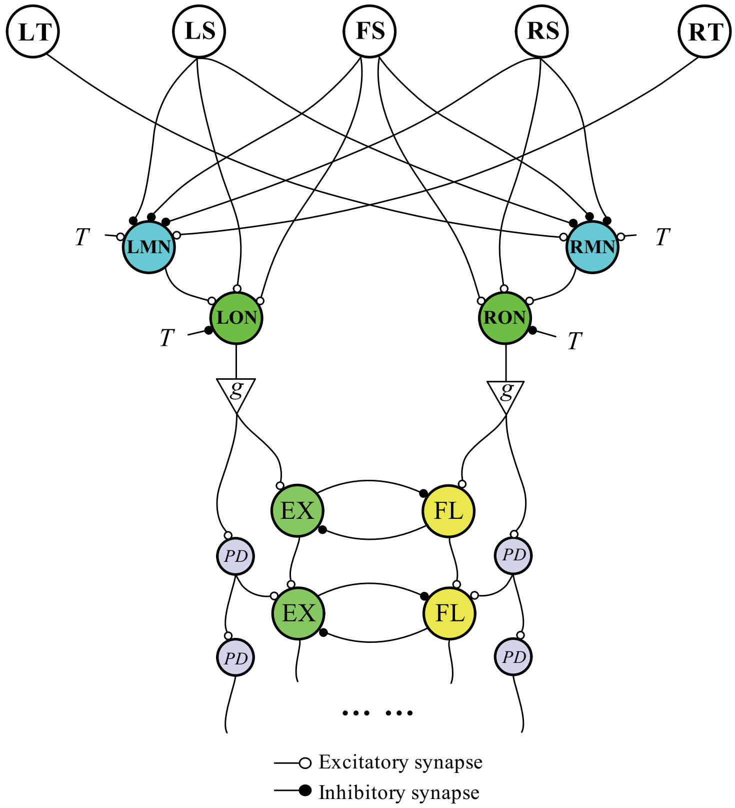


Figure 5.24. Structure of the Neural Preprocessing Network. LT: Left target command; RT: Right target command; LS: Left sensory signal; FS: Front sensory signal; RS: Right sensory signal; LMN and RMN: Left and right middle neuron; LON and RON: Left and right output neuron; T: Threshold values of neurons; PD: Phase delay; g: Gain coefficient

the adaptation degree, respectively; β is the adaptation coefficient; and $\sum S_{input}$ represents the input from outside;

Herein the threshold value of the neuron T is set as 1. This means that if the sum of the input for one neuron is larger than the threshold value, the neuron will be excited. Otherwise, the neuron will be inhibitory and have no output. There are two types of connection between the neurons: excitatory synapse and inhibitory synapse. Their mathematical meanings can be considered as merely the input multiplied by '+1' or '-1'.

To explain the principle of the network, we take situation 2 in Fig. 5.22 as an example. The sensory signals of the obstacle are [1,1,0]. After inputting this group of numbers into the network, two middle neurons will be inhibitory. Thus the left and right target commands are blocked. For the output neurons, only the left output neuron (LON) is excited. The

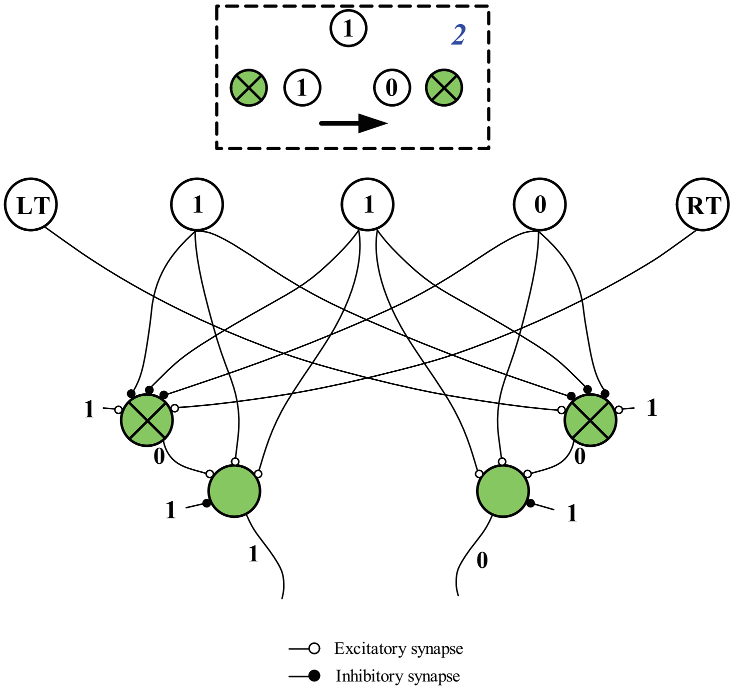


Figure 5.25. An example of the results of the neural output from the NPN

final output of the neural network is [1,0]. A chart for the results of the output from the NPN in this example is shown in Fig. ??. When the output of the LON is added to the CPG network, the left parts will have a higher driving input. This kind of asymmetric driving input will generate a bias on the oscillator output. Finally, the snake-like robot will perform turning behavior to avoid the obstacle as we expect. Similarly, if the output of the NPN is [0,1], the offset of the driving input will result in a left turning. For other situations, the results of the neural network with respect to different sensory signals are listed in table 5.2. It will have the same results as shown in Fig. 5.22 to perform a self-adaptive reactive behavior. This kind of neural network gains the advantage over traditional off-line learning methods which use complicated algorithms without online applicability. Furthermore, due to the nonlinear property of neuron model, NPN brings dynamic properties to the system that a simple logical or look-up-table operation would not have.

Table 5.2. State Table of the neurons with respect to different input

Sensory signals [LS FS RS]	Output of LMN	Output of RMN	Reactive Behavior
[0 1 0]	RT	LT	Respond to both targets
[1 1 0]	1	0	Turning behavior (right)
[0 1 1]	0	1	Turning behavior (left)
[1 1 1]	1	1	Go backward
[0 0 0]	RT	LT	Respond to both targets
[1 0 0]	RT	0	Respond to right target
[0 0 1]	0	LT	Respond to left target
[1 0 1]	0	0	Go forward

5.3.4 Preprocessing Neuron for Sensory Signals

Owing to the large noise of the IR range sensors, a preprocessing of the signals to scale them into usable values is necessary. The sensory data requires digital filtering to shape the signals as the activation of an appropriate reactive behavior. From the above analysis, the input of the NPN is either a high or a low value, which indicates the obstacle information. From this point, a sigmoidal transfer function is introduced to scale the inputs and limit the dynamic range of the sensory signals [54]. The neural model for preprocessing sensory signals can be expressed as

$$\begin{aligned} O(t+1) &= af(O(t)) + b + cS \\ S_{scaling} &= f(O(t)) = \frac{1}{1+e^{-O(t)}} \end{aligned} \tag{5.14}$$

where a represents a self-connection weight; b stands for the threshold of neural model; c is a magnification coefficient; S is the sensory signal which needs to be scaled; $O(t)$ is the time-varying intermediate variable; $S_{scaling}$ is the finally output. The structure of the model is shown in Fig. 5.26 (a).

The output voltage of the IR sensory signal is mapped onto the interval between 0 and 5. After preprocessing, the signals can be scaled into desired high and low values. An example of the sensory signals after scaling is shown in Fig. 5.26 (b). Such a filtering technique will have higher robustness with respect to a basic threshold method.

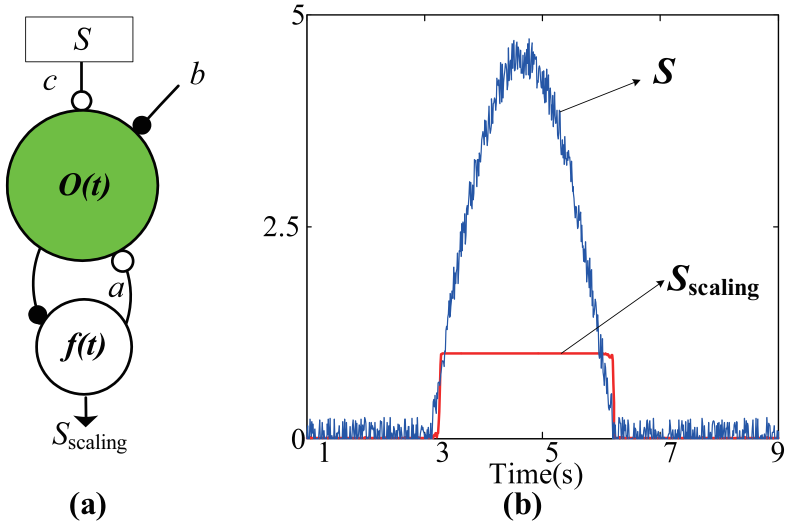


Figure 5.26. Preprocessing Neuron for Sensory Signals. (a) Structure of the scaling model; (b) Output of waves after preprocessing; a=7.2;b=6;c=6

5.4 Experimental Analysis

5.4.1 Autonomous Reactive Behavior

The proposed neural network controller will be verified by conducting an autonomous collision-free behavior both in simulation platform and a real robot *SR*-I. Firstly, the head-navigated locomotion is employed as the basic motion pattern. The arrangement of the sensors is following that in Fig. 5.21. Secondly, based on the sensory signals, the necessary turned angle Φ for the obstacle avoidance is calculated. Finally, the desired turned angle Φ can be achieved by the change of the joint amplitude corresponding to the AMM-based turning model in equation (5.11). Thus the gain coefficient g in Fig. 5.24 can be calculated by equation (5.15).

$$g = \Delta u_0 = \frac{N\pi}{n\eta}\Phi \tag{5.15}$$

The adjustment of CPG driving input will transmit one by one from the head segment to the rear segment with a fixed shift interval. The shift interval is decided by the phase delay (PD) between each neural oscillator. This kind of parameter transmitting principle gains the advantage of real-time control and smooth gait transition.

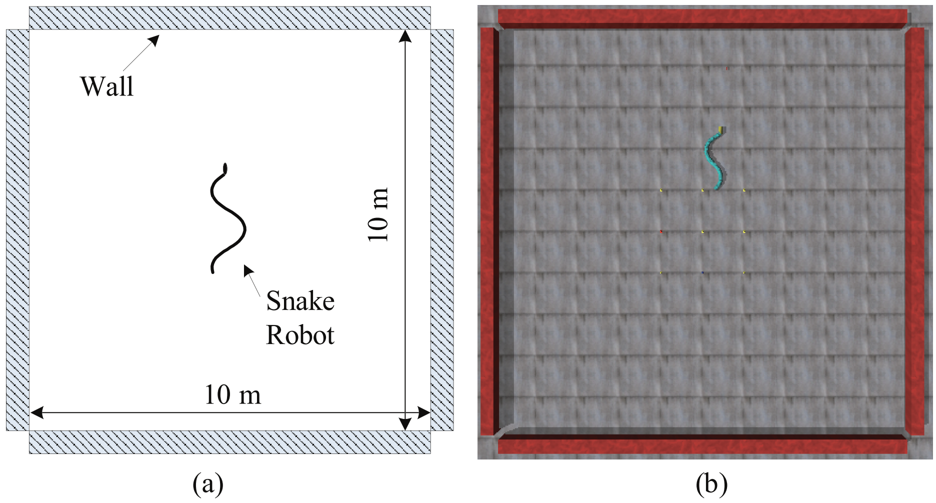


Figure 5.27. (a) Schematic of simulation environment for the collision-free behavior. (b) Screenshot of the simulation

Table 5.3. Physical parameters of the simulated robot

Numbers of segments :	Num=10
Length of link:	L_{link}=0.13m
Radius of link:	R_{link}=0.02m
Width of wheel:	L_{wheel}=0.01m
Radius of wheel:	R_{wheel}=0.03m
Weight of one segment:	M_{wheel}=0.28kg
Range of yaw angle:	[-90, +90][deg]
Friction coefficients:	μ_N=0.5 μ_T=0.02

5.4.2 Simulation Result

Based on the ODE environment, a square cloister with walls in four sides has been builded for simulation. The length of each side has 10 meter. A schematic of the designed environment is shown in Fig. 5.27 (a). Physical parameters of the simulated robot are set as those in table 5.3. The CPG parameters of the controller are defined as: $u_0 = 5.0, \tau_1 = 1.0, \tau_2 = 2.0, \beta = 2.5, w_{cpg} = 0.1, w_{neu} = 1.5, \eta = 0.4$. Initially the snake robot will be placed in the middle of the environment with random orientation.

By the use of head-navigated motion pattern, the trigger of the collision-free behavior is controlled by the distance between the wall and head. To have enough time for avoidance

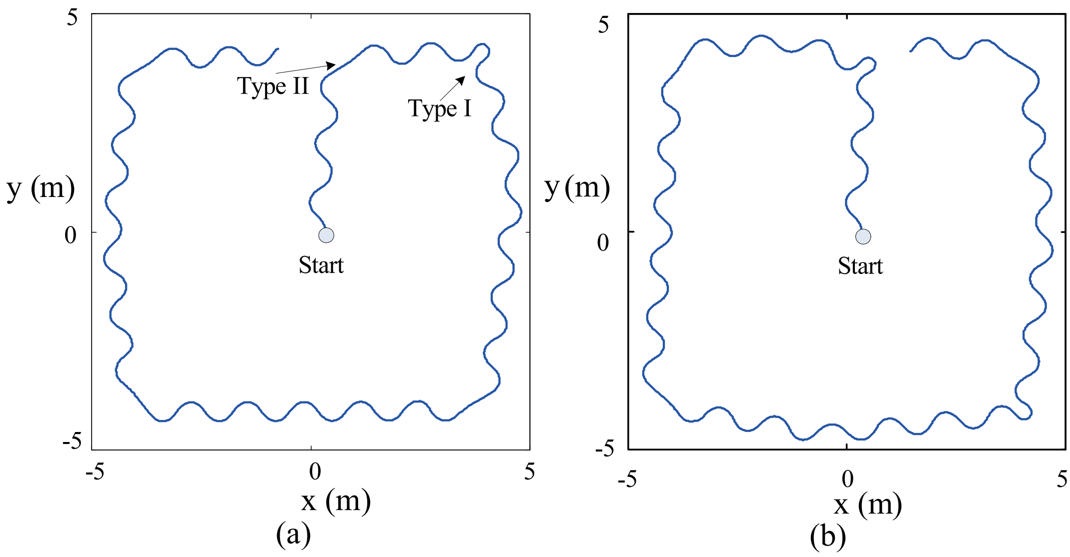


Figure 5.28. Trajectory of the simulation results. (a) Moving in clockwise. (b) Moving in counterclockwise

behavior, the distance between the wall and head for trigger point should be larger than the axial distance of one S-shape. Herein the trigger distance between the wall and the head is set as 0.7m. The proposed sensor-based neural controller is employed to process the obstacle information and activate corresponding behavior. Furthermore, a virtual target command is added to let the robot know which direction it should choose while comes with a obstacle. For simplicity, a right and left target command is used for clockwise and counterclockwise motion, respectively. The trajectories of motion in the simulation are shown in Fig. 5.28 (a) and (b). From the simulated results, the snake robot has successfully performed an autonomous collision-free behavior by the proposed method.

5.4.3 Experimental Result

Besides the simulation, the collision-free locomotion of the snake robot SR-I was conducted by the proposed sensor-driven neural controller. Fig. 5.29 shows the experimental scenarios of the snake robot SR-I which has 10 segments. During the experiment, the first joint is used for the head-navigated motion, which maintains its direction as the forward direction. On the head module, three IR range sensors (SHARP GP2Y0A710K0F) are mounted with specific angles (30deg) to detect the obstacles.

The obstacles used in the experiment are flat walls and desks which are easy to be detected by IR range sensors. During the collision-free locomotion, a virtual right target signal will be added on the controller to determine its turning direction. Initially, the snake robot will be placed in the left side with an arbitrary inclination angle. By implementing rhythmic outputs derived from the NON onto the driving joints, the snake robot will move forward and approach the obstacles. When the distance between the sensors and the obstacle come to a threshold value (here set as 35 cm), the NPN is stimulated. After the processing of the sensory signals, the NPN generates the adjusted values to drive the CPG network, and a desired turning motion for the collision-free behavior is achieved.

Due to the mechanical limitations of each joint, the largest turned angle of the robot in one period is constrained to $40°$. The turned angle in one period means the change of the direction after single turn motion, where the angle of joint is changed from A to C in Fig. 5.16. In Fig. 5.29, the snake robot performed the turning motion with three times to avoid the corner. The head joint is controlled to maintain the same direction as the advanced direction. Thus at the beginning of each turning motion, the head joint has a sudden change to adjust to the new direction. Compared the experimental results with those obtained by simulation, the largest turned angle of the snake robot is both influenced by the swinging range of each joint. However, in the simulation, the turned angle can reach $90°$ in one period. In the experiment of the real robot, if the desired turned angle is too large, the actual result is always wrong due to the slippage of wheels. Only if the desired turned angle is smaller than 40 degree, the robot can perform turn motion successfully.

5.5 Summary

In this chapter, adaptive motion control of the snake-like robot has been analyzed. Firstly, turn motion, round motion and head-navigated motion have been introduced respectively. After that, a detail modeling of the AMM-based turn motion was concluded.

Based on these results, a sensor-based neural controller was proposed to self-adaptive reactive behavior of a snake-like robot. The complete controller was designed as a topological structure composed of two main parts: neural oscillator network (NON) and neural preprocessing network (NPN). The CPG-based oscillator network was constructed as a rhythmic motion generator with nonlinear dynamics. Meanwhile, the neural preprocessing networks can fuse external signals to generate driving input for CPG network and activate an appropriate reactive behavior.

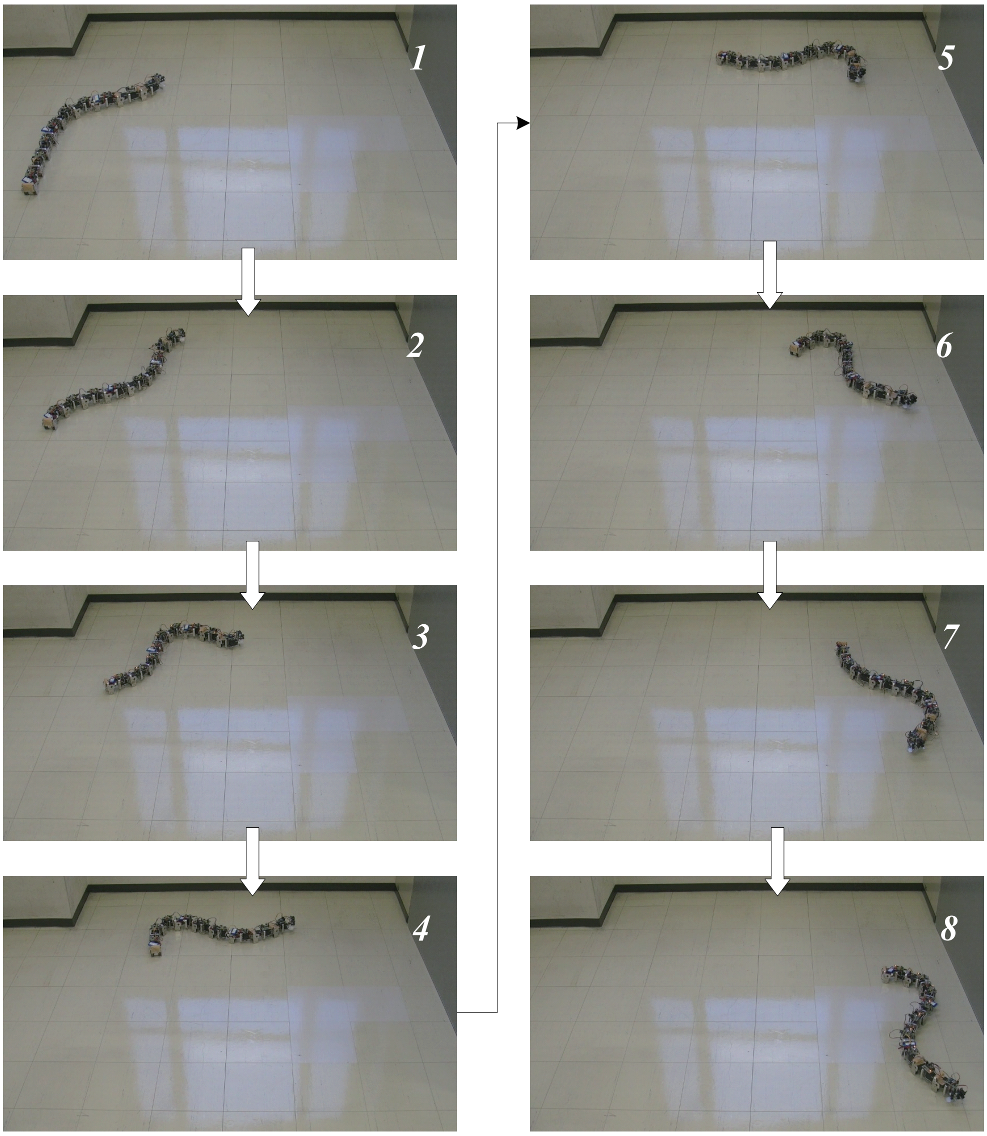


Figure 5.29. Experiment scenes of the collision-free behavior

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Due to the high degrees of freedom in snake-like robots, it is a challenge to realize effective locomotion control similar to that seen in natural snakes. To solve this difficulty, a bio-inspired central pattern generator (CPG) controller was proposed. The proposed CPG-based neural oscillator can generate self-induced oscillation, and imitate rhythmic motion pattern of animals by use of entrainment caused by inter-CPG interaction and sensory input from musculoskeletal system. Compared with the traditional methods, this kind of CPG-based controller has shown significant advantages on robustness, real-time and adaptive control.

Based on the analysis of the CPG model, the mathematical relationship between the parameters and rhythmic output were investigated. A desired serpentine locomotion of the snake robot can be performed by the proposed CPG-based oscillator network. This novel CPG network with a feedback connection does not necessitate to take additional adjustments of the CPG output due to its uniform outputs with the same amplitude and specific phase difference. Simulation and physical experiments have shown the effectiveness of the motion control by the parameter modulation in the CPG network. By utilizing the energy-based motion optimization, an optimal creeping locomotion of the robot under this controller is also obtained.

Environment sensing is a requirement for efficient snake-like robot locomotion in unknown and unstructured environments. With a suitable description of how the environment interaction affects the behavior of the snake robot, the control action in a given environment is analytically derived. Based on the results, a sensor-driven neural controller with neural oscillator network (NON) and neural preprocessing network (NPN) is designed for the self-adaptive motion of the snake robot. Such controller can couple sensory information to the corresponding body action, without necessarily the intervention of elaborate world model. The modeling of AMM-based turn motion has been proposed for the collision-free behavior of the snake robot. By fusing the peripheral sensory signals through a proposed neural controller, a driving input will be generated onto each joint oscillator to obtain a modified body configuration and perform a desired locomotion. The performance of the sensor-driven neural controller was verified by conducting an experiment on the snake robot *SR*-I.

6.2 Future work

During the study of the snake-like robot in future, many challenges still need to be solved in the design of mechanism and control system. As we know, the propulsion of the snake robot is based on forward gliding motion similar to the serpentine motion of biological snakes. The passive wheels are used to get an asymmetric friction model between the robot and the ground. This kind of mechanism has limited the adaptive motion of the snake robot in the irregular terrains where the passive wheels can not rotate well. Some studies introduced snake robot with active propulsion on the wheels in [3] and with the skin drive mechanism in [44]. The mechanical complexity of the robot with active propulsion is significantly increased. Furthermore, a snake-like robot with a passive and smooth contact mechanism can glide forward like a real biological snake. The development of a novel mechanism with the property of the snake skin would increase the motion ability significantly.

Another problem is the large degrees of freedom in the snake robot. In this study, a CPG-based neural controller is introduced for the control of the snake robot. This bio-inspired CPG-based oscillator network makes the complex dynamic of snake robot be analyzed conveniently in a control design perspective. Experimental results that demonstrate serpentine locomotion of planar snake robot with sensory information and neural controller have been presented. However, to our best knowledge, the fact still remains that non-planar locomotion in unstructured environments based on environment sensing and body shape adaptation has not yet been well studied. Therefore the future applications of snake-like robots require significantly more research on three dimensional behaviors with the help of the neural network.

Self-adaptive locomotion is a very important behavior in the control of a bio-robot. In our study, a collision-free behavior of the snake robot by the sensor-based neural controller is performed successfully. It would be useful to introduce the sensors to estimate the slope, the friction of the surface, and even more information into the proposed controller for efficient locomotion in unknown and unstructured environments. These sensors can work as low-level peripheral receptors in neural controller and be considered as local approaches in adaptive motion control. By coupling with the high-level motion command (vision, mapping, etc), global approaches of environment adaption to obtain versatile reactive behaviors would be an interesting study in future. In particular, situation where the adjustment of motion with a local approach may be prevented by a global approach is a significant challenge in the design of the controller.

Bibliography

[1] R. C. Arkin. *Behavior-Based Robotics.* MIT Press, 1998.

[2] Z. Y. Bayraktaroglu. Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot. *Mechanism and Machine Theory*, 44, Iss. 3:591—602, 2008.

[3] J. Borenstein and A. Borrell. The omnitread ot-4 serpentine robot. In *Proc. of 2008 IEEE International Conference on Robotics and Automation*, pages 1766—1767, Pasadena, CA, USA, 2008.

[4] L. Chen, Y. Wang, S. Ma, and B. Li. Studies on lateral rolling locomotion of a snake robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 5070—5074, New Orleans, LA, USA, 2004.

[5] A. H. Cohen, P. J. Holmes, and R. Rand. The nature of coupling between segmented oscillations and the lamprey spinal generator for locomotion: a mathematical model. *The Journal of Mathematical Biology*, 13:345—369, 1982.

[6] J. Collins and S. Richmond. Hard-wired central pattern generators for quadrupedal locomotion. *Biological Cybernetics*, 71:375—385, 1994.

[7] A. Crespi and A. J. Ijspeert. Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Transactions on Robotics*, 24, no. 1:75—87, 2008.

[8] F. Delcomyn. Neural basis for rhythmic behaviour in animals. *Science*, 210:492—498, 1980.

[9] K. J. Dowling. *Limbless locomotion.Learning to crawl with a snake robot.* PhD thesis, Carnegie Mellon University, 1997.

[10] O. Ekeberg. A combined neuronal and mechanical model of fish swimming. *Biological Cybernetics*, 69, no.5-6:363—374, 1993.

[11] G. Endo, K. Togowa, and S. Hirose. Study on self-contained and terrain adaptive active cord mechanism. In *Proc. 1999 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 1399—1405, Kyongju, Korea, 1999.

[12] M. Frasca, P. Arena, and L. Fortuna. *Bio-inspired Emergent Control of Locomotion Systems.* World Science, 2004.

[13] Y. Fukuoka, H. Kimura, and A. H. Cohen. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *The International Journal of Robotics Research*, 22, no.3-4:187—202, 2003.

[14] M. Golubitsky, I. Stewart, P. L. Buono, and J. J. Collins. Symmetry in locomotor central pattern generators and animal gaits. *Nature*, 401:693—695, 1999.

[15] J. Gray. The mechanism of locomotion in snakes. *The Journal of Experimental Biology*, 23:101—120, 1946.

[16] J. Gray. *Animal locomotion.* Norton, 1968.

[17] S. Grillner. *Feedback and Motor Control in Invertebrates and Vertebrates.* Croom Helm, 1985.

[18] R. Heliot and B. Espiau. Multisensor input for cpg-based sensorymotor coordination. *IEEE Transactions on Robotics*, 24, no. 1:191—195, 2008.

[19] S. Hirose. *Biologically Inspired Robot: Snake-like Locomotors and Manipulators.* Oxford University, 1993.

[20] S. Hirose and E. Fukushima. Snakes and strings: new robotic components for rescue operations. *Intl J. of Robotics Research*, 23, no. 4-5:341—349, 2010.

[21] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21, no.4:642—653, 2008.

[22] A. J. Ijspeert, A. Crespi, and J. M. Cabelguen. Simulation and robotics studies of salamander locomotion: Applying neurobiological principles to the control of locomotion in robots. *NeuroInformatics*, 3, no.3:171—196, 2005.

[23] A. J. Ijspeert and J. Kodjabachian. Evolution and development of a central pattern generator for the swimming of a lamprey. *Artificial Life*, 5, no.3:247—269, 1999.

[24] K. Inoue, S. Ma, and J. Cheng. Neural oscillator network-based controller for meandering locomotion of snake-like robot. In *Proc. of 2004 IEEE international conference on robotics and automation*, pages 5064—5069, New Orleans, USA., 2004.

[25] K. Ito and Y. Fukumori. Autonomous control of a snake-like robot utilizing passive mechanism. In *Proc. 2006 IEEE Intl. Conf. on Robotics and Automation*, pages 381—386, Orlando, USA, 2006.

[26] F. Matsuno K. Hoshino, M. Tanaka. Optimal shape of a snake robot for jumping. In *Proc. 2010 IEEE Intl. Conf. on Robotics and Automation*, pages 697—702, Anchorage, Alaska, USA, 2010.

[27] T. Kamegawa, T. Yarnasaki, H. Igarashi, and F. Matsuno. Development of the snake-like rescue robot "kohga". In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 5081—5086, New Orleans, LA, USA, 2004.

[28] H. Kimura, S. Akiyama, and K. Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillators. *Autonomous Robots*, 7, no.3:247—258, 1999.

[29] H. Kimura, Y. Fukuoka, and K. Konaga. Adaptive dynamic walking of a quadruped robot by using neural system model. *Advanced Robotics*, 15, no. 8:859—876, 2001.

[30] N. Kopell, G. B. Ermentrout, and T. L. Williams. On chains of oscillators forced at one end. *Journal of Applied Mathematics*, 51, no.5:1397—1417, 1991.

[31] Y. Kuramoto. *The handbook of brain theory and neural networks.* MIT Press, 2003.

[32] A. Kuwada, S. Wakimoto, K. Suzumori, and Y. Adomi. Automatic pipe negotiation control for snake-like robot. In *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pages 558—563, 2008.

[33] Z. Lu, S. Ma, B. Li, and Y. Wang. Serpentine locomotion of a snake-like robot controlled by cyclic inhibitory cpg model. In *Proc. of 2005 IEEE/RSJ international conference on intelligent robots and systems*, pages 3019—3024, Edmonton, Canada, 2005.

[34] Z. Lu, S. Ma, B. Li, and Y. Wang. 3d locomotion of a snake-like robot controlled by cyclic inhibitory cpg model. In *Proc. of 2006 international conference on intelligent robots and systems*, pages 3897—3902, Beijing, China, 2006.

[35] and S. Hirose M. Mori. Three-dimensional serpentine motion and lateral rolling by active cord mechanism acmr3. In *Proc. of 2002 IEEE/RSJ international conference on intelligent robots and systems*, pages 829—834, EPFL, Switzerland, 2002.

[36] S. Ma. Analysis of creeping locomotion of a snake-like robot. *Adv. Robotics*, 15, no.2:205—224, 2001.

[37] S. Ma and N. Tadokoro. Analysis of creeping locomotion of a snake-like robot on a slope. *Autonomous Robots*, 20, no.1:15—23, 2006.

[38] F. Matsuno and K. Suenaga. Control of redundant 3d snake robot based on kinematic model. In *in Proc. 2003 IEEE Int. Conf. Robotics and Automation*, pages 2061—2066, Taipei, China, 2003.

[39] T. Matsuo, T. Yokoyama, and K. Ishii. Development of neural oscillator based motion control system and applied to snake-like robot. In *Proc. of 2007 IEEE/RSJ international conference on intelligent robots and systems*, pages 3697—3702, San Diego, USA, 2007.

[40] K. Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological Cybernetics*, 52, no. 6:367—376, 1985.

[41] K. Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56, no. 5:345—353, 1987.

[42] F. Mattia, A. Paolo, and F. Luigi. *Bio-inspired emergent control of locomotion systems.* World Scientific, 2004.

[43] K. McIsaac and J. Ostrowski. Motion planning for anguilliform locomotion. *IEEE Transactions on Robotics and Automation*, 19, no.4:637—652, 2003.

[44] J. C. McKenna, D. J. Anhalt, F. M. Bronson, H. B. Brown, M. Schwerin, E. Shammas, and H. Choset. Toroidal skin drive for snake robot locomotion. In *Proc. 2008 IEEE International Conference on Robotics and Automation*, pages 1150—1155, Pasadena, CA, USA, 2008.

[45] Miller and Gavin. *Neurotechnology for Biomimetic Robots, Chap.Snake Robots for Search and Rescue.* MIT Press, 2002.

[46] Nilsson and Martin. Snake robot - free climbing. *IEEE Control System Magazine*, 18, no.1:21—26, 1998.

[47] T. Ohashi, H. Yamada, and S. Hirose. Loop forming snake-like robot acm-r7 and its serpenoid oval control. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 413—418, Taipei, China, 2010.

[48] H. Ohno and S. Hirose. Design of slim slime robot and its gait of locomotion. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 707—715, Maui, USA, 2001.

[49] [Online]. Avaliable: http://en.wikipedia.org/wiki/snakescales.

[50] [Online]. Avaliable: http://www.kerhuel.eu/wiki/index.php5.

[51] [Online]. Avaliable: http://www.snakeeducation.com.

[52] J. Ostrowski and J. Burdick. Gait kinematics for a serpentine robot. In *Proc. 1996 IEEE Int. Conf. Robotics and Automation*, pages 1294—1299, Minneapolis, USA, 1996.

[53] K. L. Paap, T. Christaller, and F. Kirchner. A robot snake to inspect broken buildings. In *Proc. 2000 IEEE Intl. Conf. on Intelligent Robots and Systems*, pages 2079—2082, 2000.

[54] F. Pasemann. Discrete dynamics of two neuron networks. *Open Systems and Information Dynamics*, 2, no. 2:49—66, 1993.

[55] M. Saito, M. Fukaya, and T. Iwasaki. Serpentine locomotion with robotic snakes. *IEEE Control System Magazine*, 22, no.1:64—81, 2002.

[56] G. Schoner, W. Y. Jiang, and J. A. S. Kelso. A synergetic theory of quadrupedal gaits and gait transitions. *Journal of Theoretical Biology*, 142:359—391, 1990.

[57] S. M. Secor, B.C.Jayne, and A.C. Bennett. Locomotor performance and energetic cost of sidewinding by the snake crotalus cerastes. *The Journal of Experimental Biology*, 163:1—14, 1992.

[58] M. Sfakiotakis and D. P. Tsakiris. Biomimetic centering for undulatory robots. *The International Journal of Robotics Research*, 26, no.11:1267—1282, 2007.

[59] M. Sfakiotakis and D. P. Tsakiris. Neuromuscular control of reactive behaviors for undulatory robots. *Neurocomputing*, 70, no.10-12:1907—1913, 2008.

[60] R. Smith. Open dynamics engine user guide.

[61] J. J. Socha. Gliding flight in the paradise tree snake. *Nature*, 418, no. 6898:603–604, 2002.

[62] G. L. Spina, M. Sfakiotakis, D. P. Tsakiris, A. Menciassi, and P. Dario. Polychaete-like undulatory robotic locomotion in unstructured substrates. *IEEE Transactions on Robotics*, 23, no. 6:1200—1212, 2007.

[63] G. Taga. A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics*, 78, no.1:9—17, 1998.

[64] G. Taga, Y. Yamaguchi, and H. Shimizu. Selforganized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147—159, 1991.

[65] K. Togawa, M. Mori, and S. Hirose. Study on three-dimensional active cord mechanism: Development of acm-r2. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 2242—2247, Takamatsu, Japan, 2000.

[66] A. A. Transeth, R. I. Leine, C. Glocker, and K. Y. Pettersen. 3-d snake robot motion: Nonsmooth modeling, simulations, and experiments. *IEEE Transactions on Robotics*, 24, no.2:361—376, 2008.

[67] D. P. Tsakiris, M. Sfakiotakis, A. Menciassi, G. La Spina, and P. Dario. Polychaete-like undulatory robotic locomotion. In *Proc. 2005 IEEE Int. Conf. Robotics and Automation*, pages 3029—3024, Barcelona, Spain, 2005.

[68] M. Walton, B. C. Jayne, and A. F. Bennet. The energetic cost of limbless locomotion. *Science*, 249, no. 4968:524—527, 1990.

[69] T. L. Williams. *Locomotor neural mechanisms in arthropods and vertebrates.* Manchester University Press, 1991.

[70] M. M. Williamson. Rhythmic robot arm control using oscillators. In *Proc. 1998 IEEE Intl. Conf. on Intelligent Robots and Systems*, pages 77—83, Victoria, Canada, 1998.

[71] P. Wiriyacharoensunthorn and S. Laowattana. Analysis and design of a multi-link mobile robot (serpentine). In *Proc. IEEE Int. Conf. Robotics, Intelligent Systems and Signal Processing*, pages 694—699, 2002.

[72] R. Worst and R. Linnemann. Construction and operation of a snake-like robot. In *Proc. IEEE Int. Joint Symp. Intelligence and Systems*, pages 164—169, Rockville, USA, 1996.

[73] X. Wu and S. Ma. Cpg-based control of serpentine locomotion of a snake-like robot. In *Proc. of 9th international IFAC symposium on robot control*, pages 871—876, Gifu, Japan, 2009.

[74] X. Wu and S. Ma. Cpg-controlled asymmetric locomotion of a snake-like robot for obstacle avoidance. In *Proc. 2009 IEEE Intl. Conf. on Robotics and Biomimetics*, pages 69—74, Guilin, China, 2009.

[75] X. Wu and S. Ma. Adaptive creeping locomotion of a cpg-controlled snake-like robot to environment change. *Autonomous Robots*, 28, no. 3:283—294, 2010.

[76] X. Wu and S. Ma. Autonomous collision-free behavior of a snake-like robot. In *Proc. 2010 IEEE Intl. Conf. on Robotics and Biomimetics*, pages 1490—1495, Tianjin, China, 2010.

[77] X. Wu and S. Ma. Head-navigated locomotion of a snake-like robot for its autonomous obstacle avoidance. In *Proc. 2010 IEEE Intl. Conf. on Intelligent Robots and Systems*, pages 401—406, Taipei, China, 2010.

[78] X. Wu and S. Ma. Development of a sensor-driven snake-like robot sr-i. In *Proc. 2011 IEEE Intl. Conf. on Information and Automation*, pages 157—162, Shenzhen, China, 2011.

[79] X. Wu and S. Ma. Sensor-driven neural controller for self-adaptive collision-free behavior of a snake-like robot. In *Proc. 2011 IEEE Intl. Conf. on Robotics and Automation*, pages 191—196, Shanghai, China, 2011.

[80] WL. Xu, F. Clara Fang, J. Bronlund, and J. Potgieter. Generation of rhythmic and voluntary patterns of mastication using matsuoka oscillator for a humanoid chewing robot. *Mechatronics*, 19, no.2:205—217, 2009.

[81] H. Yamada, S. Chigisaki, M. Mori, K. Takita, K. Ogami, and S. Hirose. Development of amphibious snake-like robot acm-r5. In *Proc. 36th Int. Symp. Robotics*, 2005.

[82] H. Yamada and S. Hirose. Study of a 2-dof joint for the small active cord mechanism. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 3827—3832, Kobe, Japan, 2009.

[83] H. Yamada, M. Mori, and S. Hirose. Stabilization of the head of an undulating snake-like robot. In *Proc. of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3566—3571, San Diego, USA, 2007.

[84] C. Ye, S. Ma, B. Li, and Y. Wang. Turning and side motion of snake-like robot. In *Proc. of the 2004 IEEE International Conference on Robotics and Automation*, pages 5075—5080, New Orleans, USA, 2004.

[85] M. Yim, D. G. Duff, and K. D. Roufas. Walk on the wild side. *IEEE Robotics and Automation Magazine*, 9, no.4:49—53, 2002.

[86] S. Yu, S. Ma, B. Li, and Y. Wang. An amphibious snake-like robot: Design and motion experiments on ground and in water. In *Proc. IEEE Int. Conf. Information and Automation*, pages 500—505, Zhuhai, China, 2009.

Appendix A

Control System

(1) Control System

To control the snake robot effectively, we develop a robust hierarchical control system in which there are two levels to perform different tasks. The architecture of control system is shown in Fig. A.1.

Module-level: For each module of the snake robot, there is a layer which is implemented to realize the basic posture control of the module using Micro-controller Unit (MCU). The MCU receives command signal from each other through Inter-Integrated Circuit (IIC) bus and gets the sensory signals from peripheral devices by Analog/Digital (A/D) ports. The motor of each joint is driven by the Pulse Width Modulation (PWM) signals from the MCU.

Computer-level: An on-line computer can communicate with the snake robot by 2.4GHz wireless module. This computer can not only send instruction signal to the head module of the snake robot, but also receive the feedback signals from the robot sensors. With the help of the wireless communication, the control system of the snake robot can be changed between centralized control and distributed control conveniently.

(2) Program

The program of the dsPIC microcontrollers is writing with the help of the Matlab-Simulink device driver Blockset. It is developed by Lubin Kerhuel in [50]. It is a easy tool for the use of dsPIC microcontroller. The schematic block diagram of the main program for snake robot in the environment of Simulink/Matlab is shown in Fig. A.2

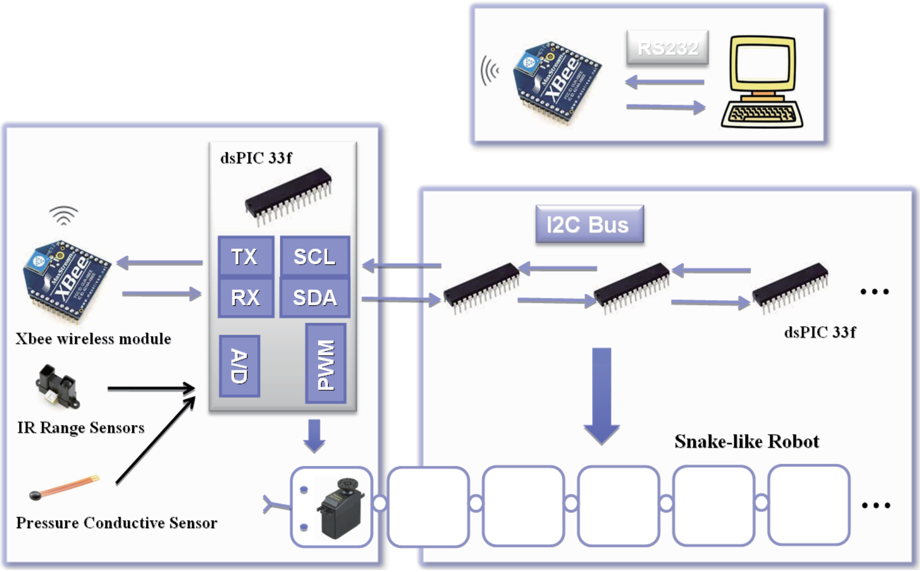


Figure A.1. Architecture of control system for the snake-like robot

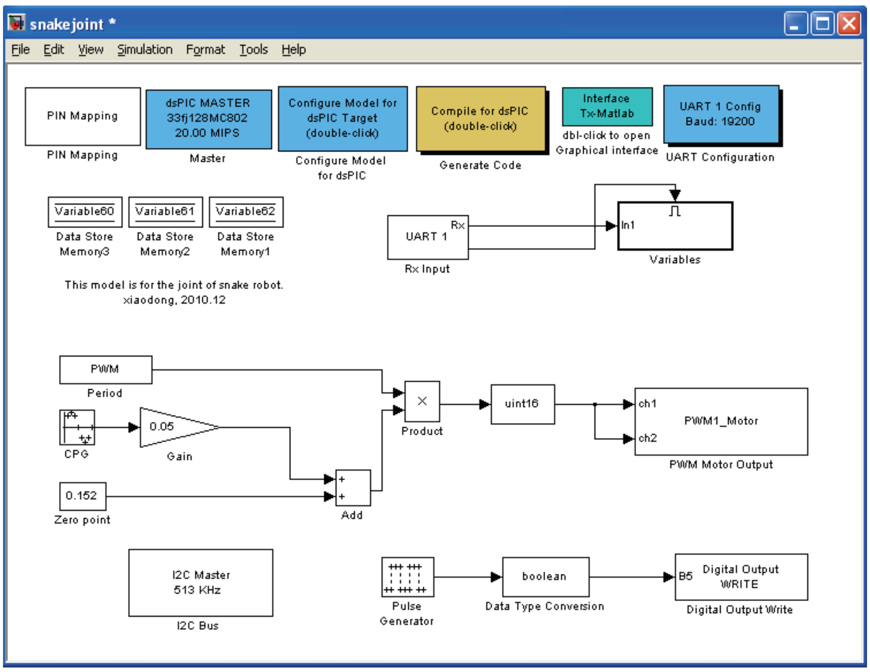


Figure A.2. Schematic block diagram of the main program in the environment of Simulink/Matlab

Appendix B

Circuit Design

A self-developed circuit board for the control of the RC motor in the joint of the snake-like robot is depicted in Fig. B.1. The left one is the head module with XBee-based wireless communication; and the right one is the joint module for motor control. Due to the limited space of the snake-like robot, the size of both circuit boards are 40 × 52 (mm). The electronic schematic and the corresponding circuit of PCB are shown in Fig. B.2 and Fig. B.3, respectively.

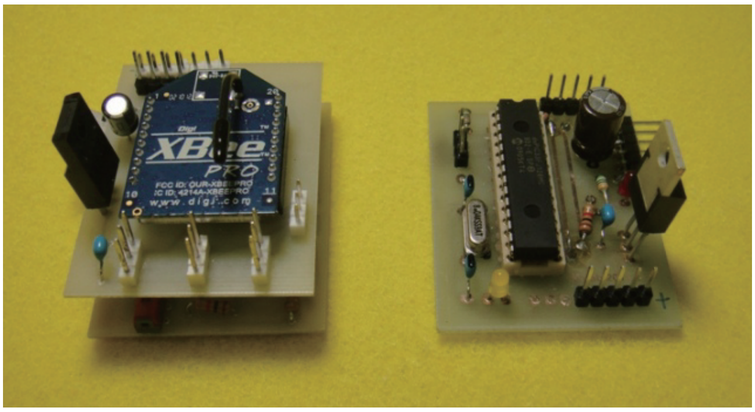


Figure B.1. The developed controller with dsPIC-based Microprogrammed Control Unit

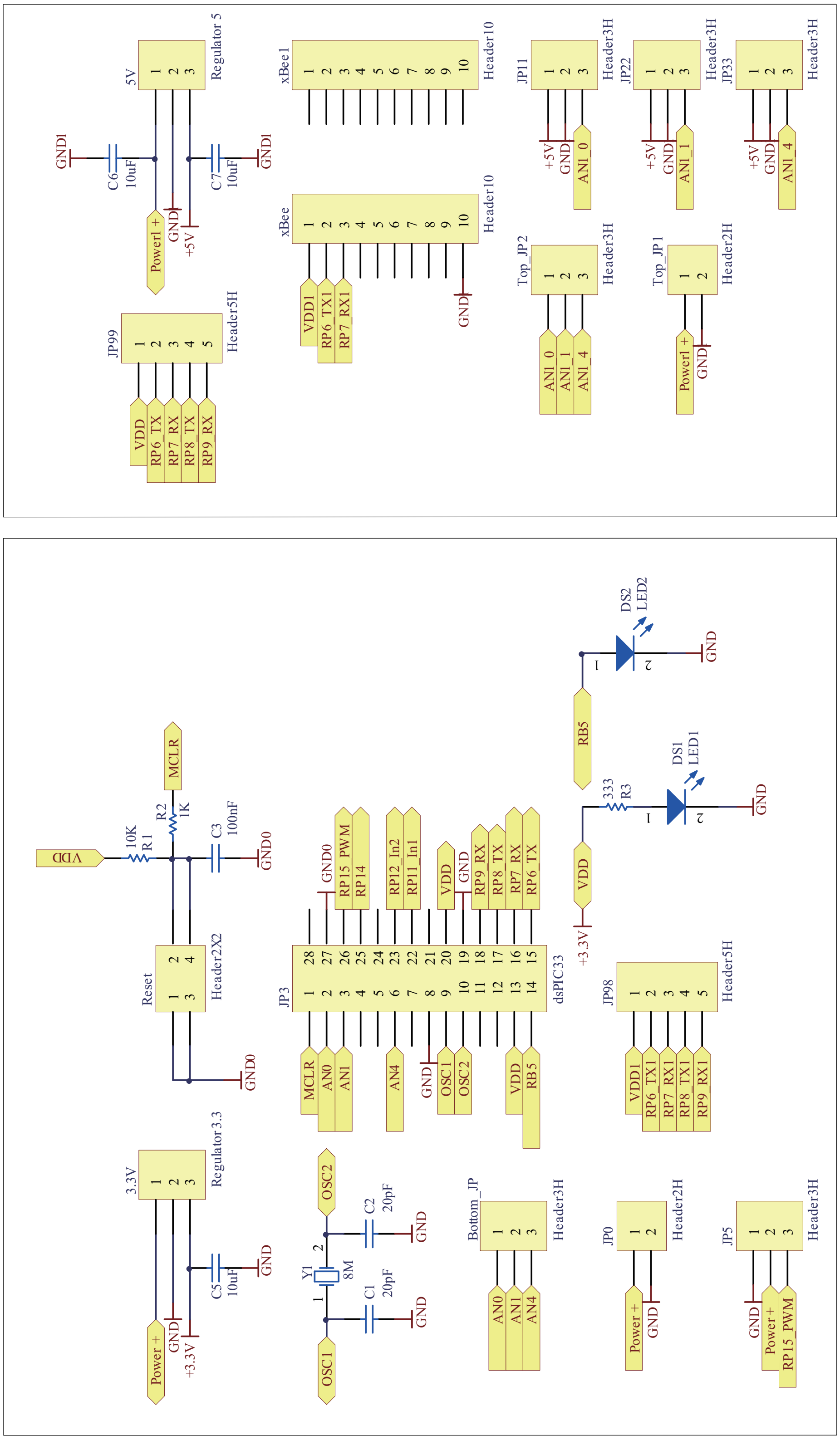


Figure B.2. Electronic schematic of the designed circuit

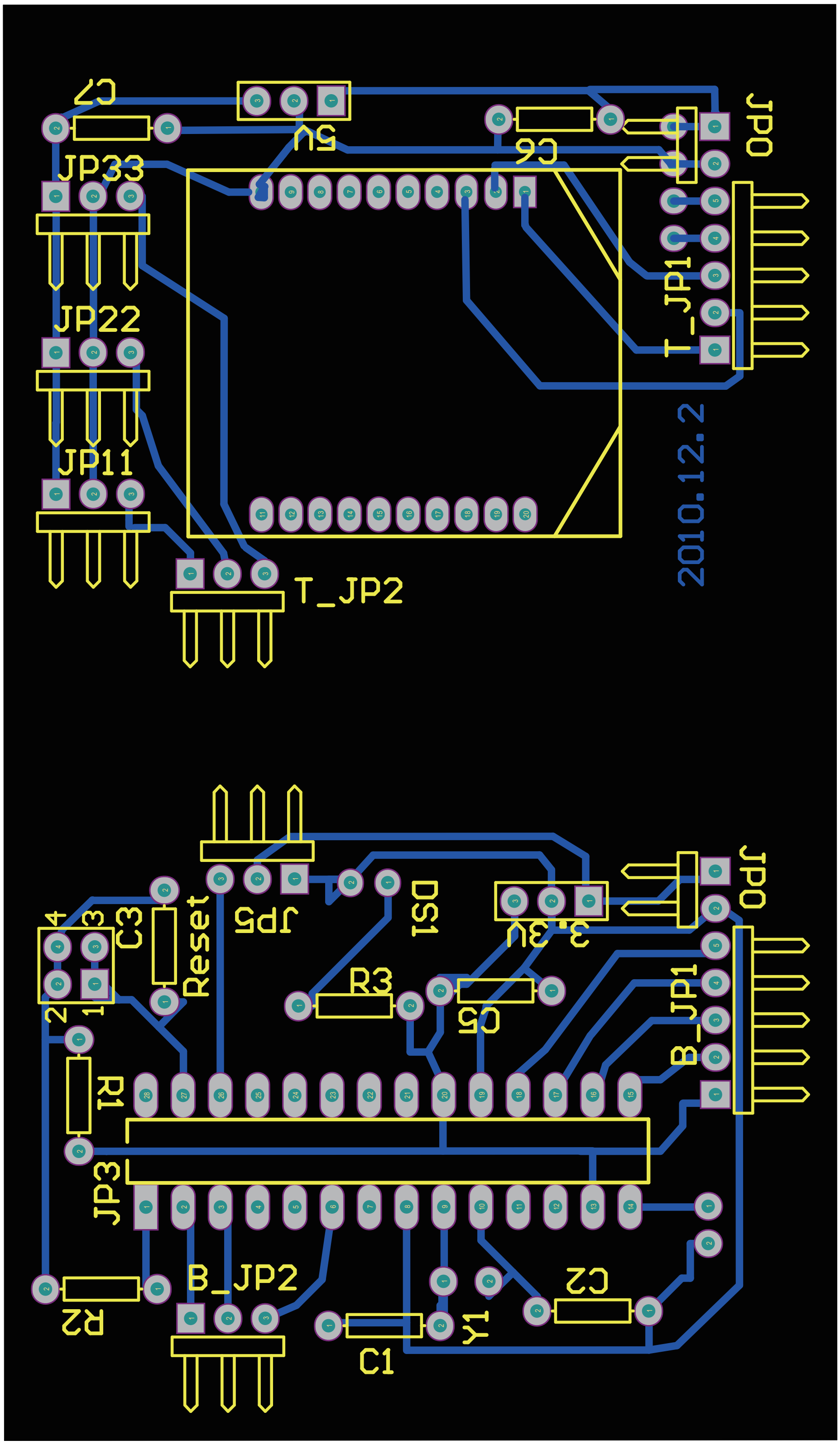


Figure B.3. Finished circuit board for each snake joint

Published Papers During Doctoral Course

Journal Papers:

1. Xiaodong Wu and Shugen Ma, CPG-based control of serpentine locomotion of a snake-like robot, ***Mechatronics***, vol.20, no.2, pp.326—334, 2010.

2. Xiaodong Wu and Shugen Ma, Adaptive creeping locomotion of a CPG-controlled snake-like robot to environment change, ***Autonomous Robots***, vol.28, no.3, pp.283—294, 2010.

Conference Papers:

1. Xiaodong Wu and Shugen Ma, CPG-based Control of Serpentine Locomotion of a Snake-like Robot, *Proceedings of the 9th International IFAC Symposium on Robot Control*(**SYROCO'09**), Gifu, Japan, pp.871—876, 2009.

2. Xiaodong Wu and Shugen Ma, CPG-controlled Asymmetric Locomotion of a Snake-like Robot for Obstacle Avoidance, *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*(**ROBIO'09**), Guilin, China, pp.69—74, 2009.

3. Xiaodong Wu and Shugen Ma, Head-navigated Locomotion of a Snake-like Robot for Its Autonomous Obstacle Avoidance, *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*(**IROS'10**), Taipei, Taiwan, pp.401—406, 2010.

4. Xiaodong Wu and Shugen Ma, Autonomous Collision-Free Behavior of a Snake-like Robot, *Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics*(**ROBIO'10**), Tianjin, China, pp.1490—1495, 2010.

5. Xiaodong Wu and Shugen Ma, Sensor-driven Neural Controller for Self-adaptive Collision-free Behavior of a Snake-like Robot, *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*(**ICRA'11**), Shanghai, China, pp.191—196, 2011.

6. Xiaodong Wu and Shugen Ma, Development of a Sensor-driven Snake-like Robot SR-I, *Proceedings of the 2011 JSME Robotics and Mechatronics Conference*(**Robomec'11**), Okayama, Japan, 2A1-K12, 2011.

7. Xiaodong Wu and Shugen Ma, Development of a Sensor-driven Snake-like Robot SR-I, *Proceedings of the 2011 IEEE International Conference on Information and Automation*(**ICIA'11**), Shenzhen, China, pp.157—162, 2011.

Books:

1. Xiaodong Wu and Shugen Ma, CPG-based control of a serpentine locomotion of a snake-like robot, in the Book ***Biologically Inspired Robotics***, Chapter 2, Edited by Yunhui Liu, Dong Sun, Taylor and Francis Group, 2011.8.