Development of Adaptive Locomotion of a Caterpillar-like Robot Based on a Sensory Feedback CPG Model

Guoyuan Li[1], Houxiang Zhang[2], Jianwei Zhang[1], Robin T. Bye[3]

[1]*Department of Computer Science, University of Hamburg, 22527, Hamburg, Germany,*

[2]*Faculty of Maritime Technology and Operations, Aalesund University College, Postboks 1517, N-6025, Aalesund, Norway. As corresponding author, telephone: +47-70161611,email: hozh@hials.no.*

[3]*Faculty of Engineering and Natural Sciences, Aalesund University College, Postboks 1517, N-6025, Aalesund, Norway*

Development of Adaptive Locomotion of a Caterpillar-like Robot Based on a Sensory Feedback CPG Model

This paper presents a novel control mechanism for generating adaptive locomotion of a caterpillar-like robot in complex terrain. Inspired by biological findings in studies of the locomotion of the lamprey, we employ sensory feedback integration for online modulation of the control parameters of a new proposed central pattern generator (CPG). This closed-loop control scheme consists of the following stages: First, touch sensor information is processed and transformed into module states. Then, reactive strategies that determine the mapping between module states and sensory inputs are generated according to an analysis of the module states. Finally, by means of a genetic algorithm (GA), adaptive locomotion is achieved by optimising the amount and speed of sensory input that is fed back to the CPG model. Incorporating the closed-loop controller in a caterpillar-like robot, both simulation and real on-site experiments are carried out. The results confirm the effectiveness of the control system, based on which the robot flexibly adapts to, and manages to crawl across the complex terrain.

Keywords: modular robot; central pattern generator; sensory feedback; adaptive locomotion

1 Introduction

Many animals exhibit highly efficient locomotion through walking, crawling, flying, and swimming. They achieve these rhythmic motions by coordination of articulation in the musculoskeletal system. Neurobiological studies of various vertebrates have shown that rhythmic movements are generated in the spinal cord by a central pattern generator (CPG). In general, a CPG consists of a group of neural networks that interact to produce coordinated rhythmic signals without necessarily requiring any sensory input [1].

In robotics, many studies have been carried out in order to understand and imitate the locomotion patterns of animals in a realistic manner, such as research involving humanoid robots [2, 3], snake-like robots [4-13], quadruped robots [14-17] and fish-like robots [18]. These robots were biologically inspired not only in mechanical

design, but also in locomotion control strategy realisation. In particular, the CPG-based control method is considered an elegant and efficient solution for online gait generation [19]. Such CPG models typically have a few control parameters for online gait modulation. Moreover, the CPG is also a kind of decentralised control method that is well suited to robots with a modular implementation. There are several famous CPG models in scientific literature, including Ekeberg's model [20], Matsuoka's model [21] and Ijspeert's model [22], all of which been investigated quite thoroughly for the past decade.

Although sensory feedback is not necessary for CPG implementation, neuroscientists have found that sensory feedback plays an important role in altering CPGs when dealing with environmental perturbations [23]. To achieve adaptive locomotion, there are several approaches for taking sensory feedback integration into account when altering the control parameters of CPG models. In [15], Kimura designed sensory feedback resulting from motions induced by reflexes to change the period of CPG model. He succeeded in realising adaptive dynamic walking of the Tekken robot on irregular terrain. Another successful example is from Ryu et al., who proposed a frequency-adaptive oscillator that can learn and adapt to changes in the frequency of sensory feedback signals [8]. The control strategy was verified by employing a snake-like robot moving with constant velocity over terrain with varying frictions. Zhu et al. proposed a CPG based controller with sensory feedback [9]. They developed feedback law on a snake-like robot and succeeded to achieve effective motion behaviour in 2D plane. Yet another example comes from Sato et al., who designed a discrepancy function to change the phase of oscillators of a snake-like robot so that the robot exhibited highly adaptive behaviour for climbing slopes [11, 12].

Caterpillars are among the most successful climbers – they can manoeuvre in complex three-dimensional environments, burrow, and hold on to substrates using a very effective passive grasping system [24]. Our DFG on-going project is focused not only on understanding the principles behind the locomotion of caterpillars but also on trying to imitate their adaptive and efficient locomotion behaviour [25]. In this paper, we propose a hierarchical control mechanism based on a sensory feedback CPG model to realise the adaptive locomotion of a caterpillar-like robot moving on irregular terrain. The contribution of this paper is threefold. First, the proposed CPG circuit provides a possible solution for easily integrating sensory feedback at the neural level. Second, a closed-loop control system is developed by incorporating sensory feedback from the environment. Through a genetic algorithm (GA), sensory information is analysed and transformed as sensory input into the CPG model, which helps the robot to adapt to the environment. Third, simulation and real on-site experiments are implemented to verify the effectiveness of the proposed control mechanism.

2 Robotic control system

Figure 1 illustrates the control architecture of the robot. It consists of three main components: The locomotion control part, the environment part, and the reaction control part.

The locomotion control component plays a role in gait generation. A CPG model inspired by the lamprey's spinal circuit is as well-developed as the controller of the robot. The output of the model is transmitted as desired joint angles to each module of the robot. Due to the phase lag between the modules, the robot can perform peristaltic crawling behaviour like real caterpillars.
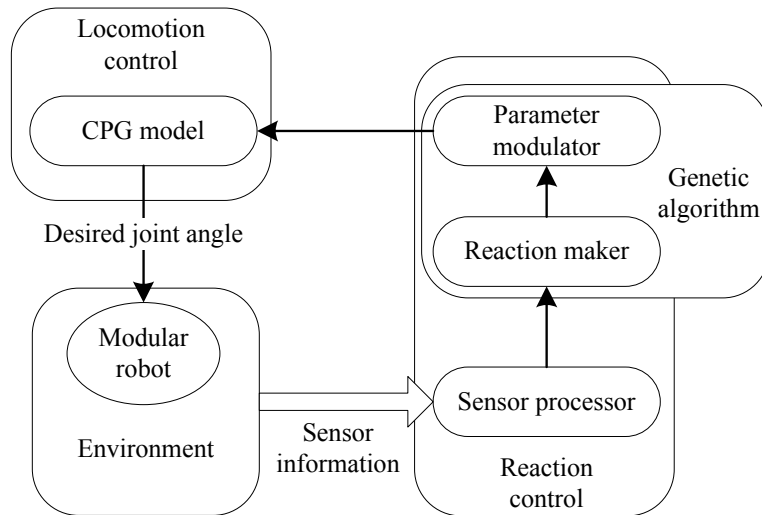
Figure 1. Adaptive control architecture.

In the environment component, sensory information is collected regularly from touch sensors on the robot. After interacting with the environment, the sensory information is transmitted to the reaction control component.

The reaction control component is the key part of the whole control system and can be further divided into three sub-functional parts, namely a sensor processor, a reaction maker, and a parameter modulator:

- **The sensor processor** filters the raw sensory information and converts it to a sequence, or string, of binary bits. Since the raw sensory information is sampled and accumulated over a period of time, each converted binary bit contains accumulated contact information instead of an instantaneous contact event (see details in Section 4.1). Thus the binary string represents the accumulated contact state of the robot, eliminating the impact of frequent change of instantaneous contact information.

- **The reaction maker** determines module reflex behaviour. For each module of the caterpillar-like robot, the reaction maker maps the contact states into a

reaction rule that determines whether to generate a sensory input to the module in a given time step.

- **The parameter modulator** is responsible for generating sensory input for the CPG model according to the reaction rules, so as to adaptively shape the locomotion gait to be compliant with the environment.

Although the control architecture described above may seem sufficient for our closed-loop control system, there are still several key issues that must be solved. First, the mapping mechanism between the contact states and reaction rules need to be considered. Second, one must determine what amount of sensory input should be generated in such a manner as to successfully achieve adaptive movement. Together, these two aspects require that a large number of parameters be simultaneously and appropriately chosen. Since this task is not easily solved with analytical methods, a GA is selected as the tool to find an optimal solution that yields satisfactory adaptive behaviour. Once the set of related parameters has evolved, the reaction maker and the parameter modulator can operate properly. As a result, the robot is endowed with the capability of traversing over complex terrain.

3 Locomotion control

Although the underlying biological details of complex neural CPG circuits in many animals are not yet fully understood, the simpler neural circuitry in the spinal cord of lampreys has been thoroughly studied [26]. In this section, a CPG circuit inspired by the neural circuit diagram of lampreys is designed as the locomotion control model of the robot. The resulting model not only features the ability to modulate CPG parameters such as amplitude, period, phase difference and offset, but also provides a solution inspired by biological findings in the lamprey for realising the sensory feedback at the

neural circuit level. The locomotion control model consists of a neural model and a chained inhibitory circuit that are discussed in the following sections, respectively. Here, we will only give a brief introduction to the new proposed CPG model in order to aid the explanation of the overall control mechanism. The details of this new CPG are presented in [27].

3.1 Neural model

As shown in Figure 2, the neural model consists of two parts: a motor control centre and a chain of oscillators connected to the motor control centre. The motor control centre is the brainstem of the biological controller. It descends motor commands through command neurons to modulate the behaviour of a command oscillator, which in turn is responsible for generating rhythmic signals that are passed to the chain of oscillators.

The circuit on the two sides of an oscillator are mirror images to one another. Rhythmic activity is generated by interaction among four types of interneurons, namely crossed interneurons (CIN), lateral interneurons (LIN), excitatory interneurons (EIN), and motoneurons (MN). The CIN projects an excitatory synapse onto the EIN and projects inhibitory synapses onto the LIN, CIN, and MN, all on the contralateral side; the EIN projects excitatory synapses onto the other three types of interneurons on the ipsilateral side of the circuit; the LIN projects an inhibitory synapse onto the ipsilateral CIN; and finally, the output of the oscillator is determined from a comparison of the two MNs on each side of the circuit. A major difference compared to the original neural diagram of the lamprey is that the synapses emitted from CINs to EINs are excitatory instead of inhibitory.

Figure 2. The neural model consists of a motor control centre with a command oscillator connected to a chain of oscillators. The motor control centre is responsible for neuromodulation, while the command oscillator is responsible for rhythmic activity generation.

In addition to the internal coupling, an oscillator can also project synapses onto other oscillators through the EINs, and receive synaptic interaction from other oscillators through LINs. The resulting dynamic system of each side of the ith oscillator can thus be described by the following equations:

$$\tau x'_{\{CIN\}i} = -x_{\{CIN\}i} + \sum \omega_s s_{\{CIN\}i} \tag{1}$$

$$\tau x'_{\{LIN\}i} = -x_{\{LIN\}i} + \sum \omega_s s_{\{LIN\}i} + \sum \omega_c c_{\{LIN\}j} \tag{2}$$

$$\tau x'_{\{MN\}i} = -x_{\{MN\}i} + \sum \omega_s s_{\{MN\}i} + \beta A \tag{3}$$

$$x_{\{EIN\}i} = \frac{A}{1+e^{x_{\{\overline{CIN}\}i}}} - \frac{1}{2}A \tag{4}$$

$$output_i = \max(x_{\{MN\}i}, 0) - \max(x_{\{\overline{MN}\}i}, 0) \tag{5}$$

where x_i is the state of each interneuron, the subscript in curly brackets indicates the type of interneuron, and its overline indicates the state of interneurons at the contralateral side of the oscillator; s_i represents synapses with a synaptic weight ω_s received from other interneurons in the same oscillator; c_j represents synapses with a synaptic weight ω_c received from other oscillators; τ, A and β are tunable parameters that determine the oscillator's period, amplitude and offset, respectively; and *output$_i$* is the oscillator's output. The sign of the synaptic weight ω_s or ω_c determines the characteristic of the corresponding synapse. A positive value represents an excitatory synapse, whereas a negative value represents an inhibitory synapse.

To ensure smoothness of the output, all of the synaptic weight parameters in an oscillator have an absolute value of 1.0, except for the synapses from EINs to MNs, whose weight parameter's absolute value is 0.1. Moreover, the initial state of the interneurons at the start of an oscillation is important, as it was found that only a small initial value of interneurons and a slight initial asymmetry between the two CINs on both sides can give rise to self-sustained oscillation.

3.2 Chained inhibitory circuit

The chained inhibitory circuit is designed for linear gait generation. Figure 3 shows the chained topology of a set of connected oscillators. In each oscillator, 'L' and 'R' represent the interneurons on the left and right side, respectively. In order to maintain a fixed phase difference, each oscillator is required to project unidirectional inhibitory synapses onto its adjacent oscillator. The synaptic weights between these oscillators are all set to *-1*.

The command oscillator that belongs to the motor control centre projects inhibitory synapses onto the chained topology. As shown in Figure 3, the command oscillator is self-connected by using two additional synapses, one excitatory and one

inhibitory. The parameters α and γ related to the two additional synaptic weights are tunable parameters for phase difference modulation and lie in the range [*0, 1*]. The parameter γ is usually regarded as a constant ($\gamma = 0.2$ in this paper) so as to simplify the control of the phase difference. This means that only the parameter α has an effect on phase difference modulation. Figure 4 illustrates the phase difference modulation with respect to parameter α. It is noted that the phase difference decreases with the growth of α. The available phase difference lies in the range [*45°, 145°*].
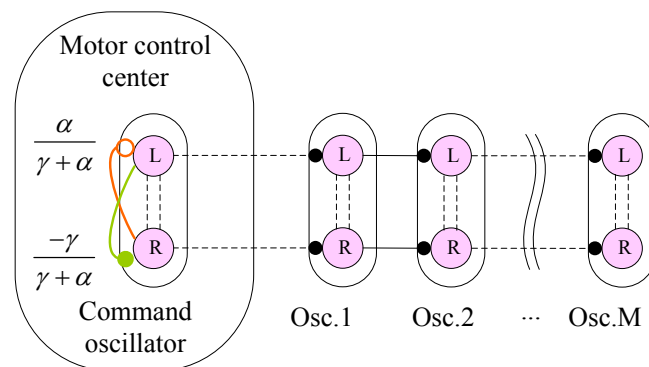
Figure 3. The chained topology of oscillators. Two additional synapses in the command oscillator determine the phase difference between these oscillators. Parameters α and γ within the range [0, 1] are used for phase difference modulation.
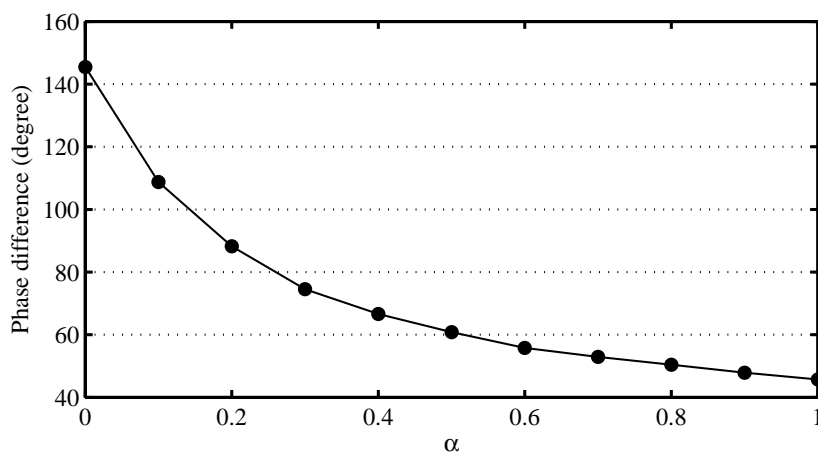

Figure 4. Phase difference modulation with respect to parameter α. The phase difference can be modulated in the range from 45° to 145°.

4 Reaction control

The reaction control component is an essential part of the control system and enables the robot to move adaptively when confronted with irregular terrain. As mentioned before, there are three sub-functional stages in the reaction control component, namely a sensor processor, a reaction maker, and a parameter modulator. In addition, there is a fourth mechanism related to motion optimisation, in which a GA is used for adaptively tuning sensory input parameters. These four stages are presented in the corresponding sections below.

4.1 Sensor processor

The sensor processor handles the raw touch sensor data and identifies the contact state for each module at regular intervals of time. Importantly, "module state" indicates accumulated contact information rather than instantaneous contact state. There are only two possible module states, namely "periodic touch" and "hanging in the air", both of which imply an accumulated contact state over one period of time. Figure 5 shows an example of how a module state is determined using only touch information.

Assume a robot performs a linear gait during the locomotion. Each module of the robot therefore moves up and down, tapping the terrain periodically. Figure 5(a) shows the raw touch sensor data of two consecutive modules *i* and *i+1* when the robot is climbing a slope, with spikes in the force measured by the touch sensors, indicating contact. The top of Figure 5(b) shows the instantaneous module states that result from applying a threshold function to Figure 5(a), whereas the bottom of Figure 5(b) shows the module states based on accumulated contact information. In this example, modules *i* and *i+1* have periodic touch for the first *2.4s* and *1.9s*, respectively, before they begin hanging in the air.
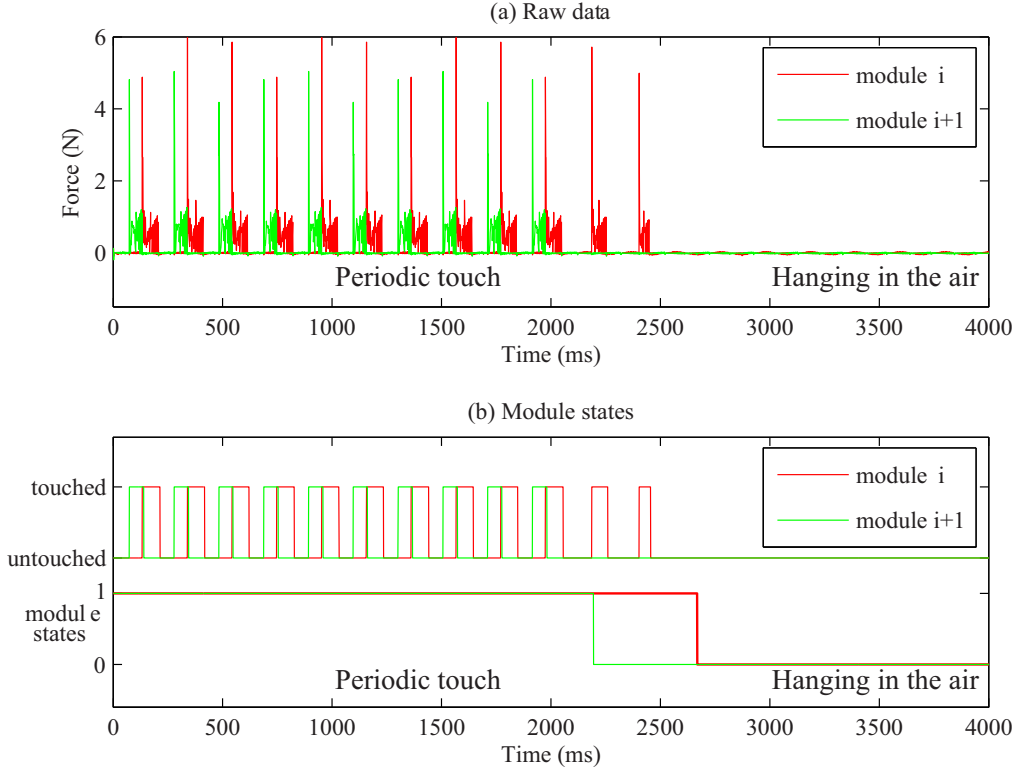
Figure 5. Sensor data processing. (a) Raw data of touch sensors are gathered when the robot is climbing a slope. (b) Module states are produced by filtering the raw data and accumulating the touch state over a period of time.

To identify the change of module states automatically, a filter is used to classify the raw data as binary values, representing either a touched or an untouched state, respectively, as seen at the top of Figure 5(b). This filter can be described as follows:

$$t_{(i,n)} = G(R_{(i,n)}) \tag{6}$$

where t denotes the instantaneous touch state of the ith module at time step n; R is the raw data of the ith module's touch sensor; and G is a threshold function.

However, because the instantaneous touch states generally change frequently, using this information for CPG modulation may lead to unstable, jerky behaviour. Instead, accumulating instantaneous values of the touch states over more than one period of locomotion as shown in (7) yields an acceptable solution for module state identification:

$$s_{(i,n)} = \begin{cases} 1 & \text{if } \sum_{n'=n-kN}^{n} t_{(i,n')} > 0 \qquad (k \geq 1) \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

where $s_{(i,n)}$ is the state of the *i*th module at the time step *n*; *N* denotes the number of time steps in one period of locomotion; and *k* is the number of historic touch states to consider. The module state value of "1" means that the module is touching the surface, whereas a "0" means that the module is in the air. Using the history of touch states to determine the module states causes a lag of approximately *(k-1)N* samples between the raw touch sensor data and the processed binary module states. In the example in Figure 5, a history of *k=1.5* is used, which means that since there are about *200ms* between each detected touch (sample), the delay in detecting a state change is about *100ms* (see bottom of Figure 5(b)).

4.2 Reaction maker

The reaction maker is used to produce reasonable reactive behaviour of the robot. It takes the module states into account and generates proper external stimuli λ that will be fed back to the CPG model (see details in Section 4.3), so as to shape the CPG output and adapt the robot to the environment. Suppose the robot's status can be described as a sequence of the module states, where each module state is represented as a binary function, as seen in (8):

$$r_n = \langle s_{(1,n)}, s_{(2,n)}, \ldots, s_{(m,n)} \rangle \qquad (8)$$

According to analyse the binary string of r_n, there are three cases that the reaction maker should identify:

- *Case 1*: The first case concerns a border reaction, where the head or the tail of the caterpillar-like robot is not touching the substrate. Suppose the caterpillar-

like robot is in a state where the tail modules are all in touch with the substrate and the head modules are not, or vice versa. This implies that r_n has a form like <0, 0...**0**, **1**...> or <...**1**, **0**, 0...0>. The reaction maker is then responsible for finding the joint from the border side where two adjacent module states are 0 and 1 (indicated with boldface). Then a certain amount of stimulus, represented as λ_1 and λ_2 for head and tail side respectively, is integrated into the corresponding oscillator of the joint, so that the border body of the robot is able to make contact on the substrate.

- *Case 2*: The second case concerns a "camelback" pose of the caterpillar-like robot. Suppose the caterpillar-like robot is in a condition where two internal modules are touching the substrate, whereas a number of modules in the middle are not. This implies that r_n has a form like <...1, 0, **0...0**, 0, 1... >. In this case, a certain amount of stimulus, represented as λ_3, is added to the corresponding oscillators of the joints whose adjacent module states are both 0 (indicated with boldface). This, in turn, causes the middle modules of the robot to adapt to the terrain.

- *Case 3*: The final case concerns the recovery of the body shape during locomotion. To prevent the body shape of the caterpillar-like robot from excessively adapting to the terrain, any afferent stimuli to oscillators should be removed once the corresponding joint of the robot has adapted to the terrain. This mechanism can help the robot to resume from a modified gait to the normal one. In this case, the reaction maker tries to identify continuous ones in the binary string r_n, such as the form <...0, 1, **1...1**, 1, 0...>. The integrated stimuli on these joints whose adjacent module states are both 1 (indicated in boldface) are

removed, so that the affected joint oscillation could gradually recover to original state.

4.3 Parameter modulator

To deal with sensory information, sensory neurons are integrated into the model, as shown in Figure 6. For simplicity, the ellipses labelled "interneurons" represent the CIN, LIN and EIN on each side of the oscillator. The sensory neurons on each side of the oscillator set excitatory and inhibitory synapses to motoneurons. Due to the mirrored functionality of these sensory neurons, the two sensory neurons can be considered together. When an external stimulus λ is transmitted to a sensory neuron, the sensory neuron will in turn have an inhibitory effect on the motoneuron on the same side and an excitatory effect on the motoneuron on the opposite side. As a result, the external stimulus will add a contribution to the output of the oscillator. We can thus rewrite (3) as follows:

$$\tau x'_{\{MN\}i} = -x_{\{MN\}i} + \sum \omega_s s_{\{MN\}i} + \beta A \pm x_{\{SN\}i} \tag{9}$$

where $x_{\{SN\}}$ represents the output of the sensory neuron and the positive or negative sign of the last term indicates an excitatory or inhibitory effect on the motoneuron, respectively.

The dynamic change of the sensory neuron is a piecewise function. It depends on what case its corresponding joint belongs to. For example, as described in the previous section, when a joint satisfies case 1 or 2, a certain amount of stimulus λ_p is accumulated in its sensory neuron, as shown in (10).

$$\delta_m x'_{\{SN\}i} = -x_{\{SN\}i} + A \cdot \sum_{n_0}^{n} \lambda_p \qquad p \in \{1,\ 2,\ 3\} \tag{10}$$
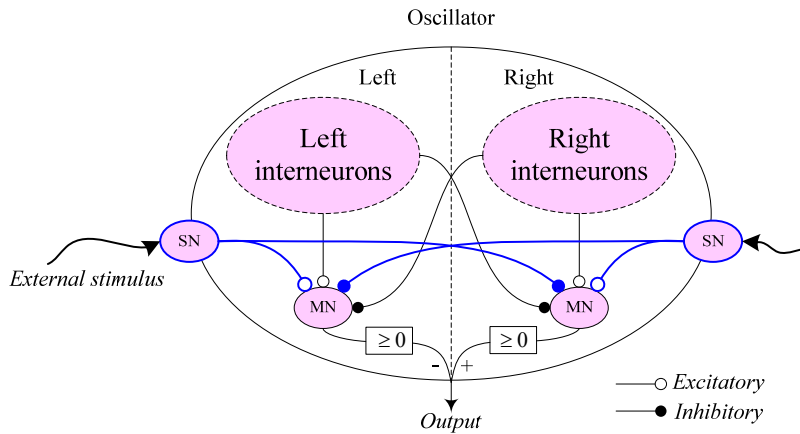
Figure 6. Sensory neuron integration. The oscillator with its interneurons lumped together is the same to the one in Figure 2. Two additional sensory neurons are employed on each side of the oscillator for sensory feedback.
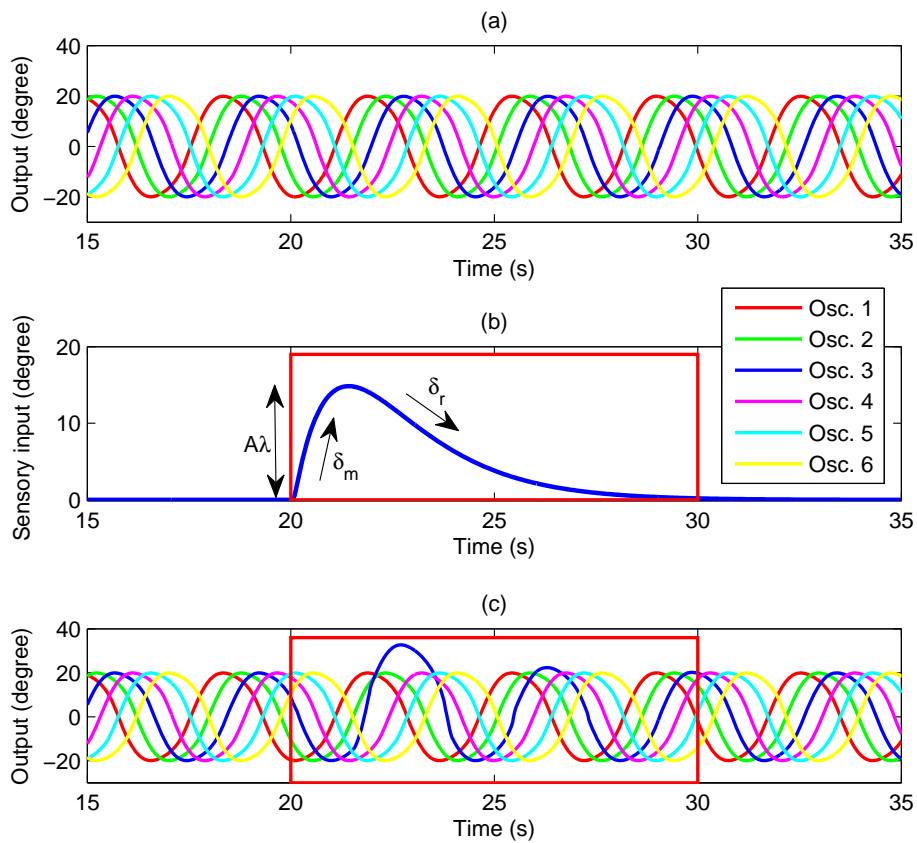

Figure 7. Effect of the sensory integration on the CPG output. (a) Normal CPG output. (b) The sensory input generated in oscillator 3 varies with the amount of afferent stimuli. (c) The variation of the CPG output after sensory integration over time.

where δ_m is a time constant that controls how quickly the output signal $x_{\{SN\}}$ will change; n is the current time step; n_0 is the time step when the state of the corresponding joint changes from case 3 to case 1 or 2; the sum of λ_p indicates an accumulation of stimulus from the time step n_0 to n; and A denotes the desired amplitude of the oscillator.

If a joint belongs to case 3, its sensory neuron with a time constant δ_r will gradually remove the effect on motoneurons, as seen in (11):

$$\delta_r x'_{\{SN\}i} = -x_{\{SN\}i} \tag{11}$$

Figure 7 explains how the sensory input affects the joint angle generated by the sensory neurons in the CPG model. Figure 7(a) shows normal joint angle generated using the chained inhibitory circuit. In Figure 7(b), some amount of accumulated stimuli is transmitted into the sensory neuron in one of the oscillators at the time *20s*, where its corresponding joint satisfies case 1 or 2. After 2 additional seconds at time *22s*, the generated sensory input gradually recovers to a value of zero since the corresponding joint switches are changed to case 3 and there are no further stimuli afferent to the oscillator. Figure 7(c) illustrates the result of sensory integration. Sensory integration is achieved by means of the superposition of the generated sensory input and the normal CPG output. Note that the goal of this example is not to emphasise the resultant coordination of all the oscillators after the integration of sensory neurons. Instead, it is only to emphasise the fact that the CPG model is able to respond to external stimuli with the help of integrated sensory neurons.

4.4 Motion optimisation

As described above, the sensory neuron integrated in the CPG model plays an important role in adaptive locomotion. However, it is difficult to determine the amount of stimulus

λ for each case, as well as the time constant (δ_m and δ_r) for the output of the sensory neuron.

In this work, genetic algorithm (GA) is employed as a solution to evolve related parameters λ, δ_m and δ_r in the CPG. A real number GA derived from the standard binary GA is used, with genes containing real values within a specified range instead of employing binary values. Candidate solutions are encoded as chromosomes of fixed-length strings of genes. Each gene corresponds directly to one parameter. For our control model, there are 5 parameters that need to be evolved in each oscillator, as shown in Table 1.

A fitness function is defined over the genetic representation and measures the quality of the represented solution, as seen in (12):

$$fitness = \eta \cdot \frac{v}{v_0} + (1-\eta) \cdot \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{s(i,j)}{mn} \tag{12}$$

where η ($0<\eta<1$) is a proportional variable; v is the average speed of the robot climbing over uneven terrain, whereas v_0 is the speed of the robot climbing on flat terrain; $s(i, j)$ is the module state; and m and n are the module number and time step, respectively.

Table 1. Optimised parameters of CPG model.

Parameters	Case	Description	Range
λ_1	1	Stimulus for head bending	[-1, 1]
λ_2	1	Stimulus for tail bending	[-1, 1]
λ_3	2	Stimulus for internal bending	[-1, 1]
δ_m	1 or 2	Time constant for modulation	(0, 20]
δ_r	3	Time constant for recovery	(0, 20]

The aim of the GA algorithm is to find the best chromosome that enables the fitness function to reach a maximum value. The fitness function in (12) is a weighted sum of two different components of locomotion. The first component rewards the velocity of the adaptive locomotion for the robot climbing over uneven terrain relative to the velocity over flat terrain. The second component rewards locomotive stability measured as the percentage of the average number of modules that are in contact with the terrain. The weighting factor η in (12) determines which component has a higher priority during the evolutionary process and ensures that constrains the fitness function is to the range (0, 1). In this study, η is set to *0.7*, which means that the locomotive speed in uneven terrain is weighted more than locomotive stability.

5 Simulation and experiment

Simulation and on-site experiment were carried out to validate the effectiveness of the control system in realising adaptive locomotion with sensory feedback. To imitate the motion of real caterpillars in this experiment, we used a linear gait and followed the investigation of the locomotive parameters in [28]. Thus, all oscillators in the chained inhibitory circuit are predefined, with an amplitude of *20°* and a phase difference of *120°*.

5.1 Simulation

For the simulation, a simplified modular caterpillar-like robotic configuration was designed for the locomotion research in ODE [29]. Each mechanical module of the robot is designed as a simple rigid box. A one-dimensional joint rotating along the horizontal axis, located between every two modules, allowed adjacent modules to rotate in a vertical plane with a range of ±90°. Touch sensors were installed on the abdominal parts of the robot, with a function similar to that of the prolegs of living caterpillars. As

a result, the control system of the robot is able to collect terrain contact information regularly. The related parameters of the robot module are listed in Table 2.

Table 2. Specification of the simulated robot module.

Components	Module	Touch sensor
Length (mm)	72	10
Width (mm)	52	52
Height (mm)	52	10
Weight (g)	150	10

Table 3. GA operations and parameters.

Genome type	Real genome
Selection	Roulette selection and elite selection
Crossover	Uniform crossover
Mutation	Random number
Population size	100
Crossover rate	0.9
Mutation rate	0.01
Termination criterion	Converge to 0.99 after 50 generations

A simulation scenario is also constructed, consisting of ramps with different inclinations and different angles between them, as shown in Figure 8(a). The goal is to make the robot climb over the complex terrain adaptively and autonomously.

First, the GA algorithm implemented in GAlib library [30] was utilised to evolve CPG parameters listed in Table 1 to optimise the robot's climbing skills. Details of the GA operations and parameters applied in this experiment are listed in Table 3. During the GA progress, each simulation was run for a fixed number of time steps and an

evaluated value was returned based on the fitness function. After the evolution of the GA, the best chromosome is chosen as the evolved result of the CPG parameters. Then, according to these evolved parameters, the robot is controlled by the following routine at each time step n:

(1) Get raw sensor data at time step n;

(2) Process the data and update r_n;
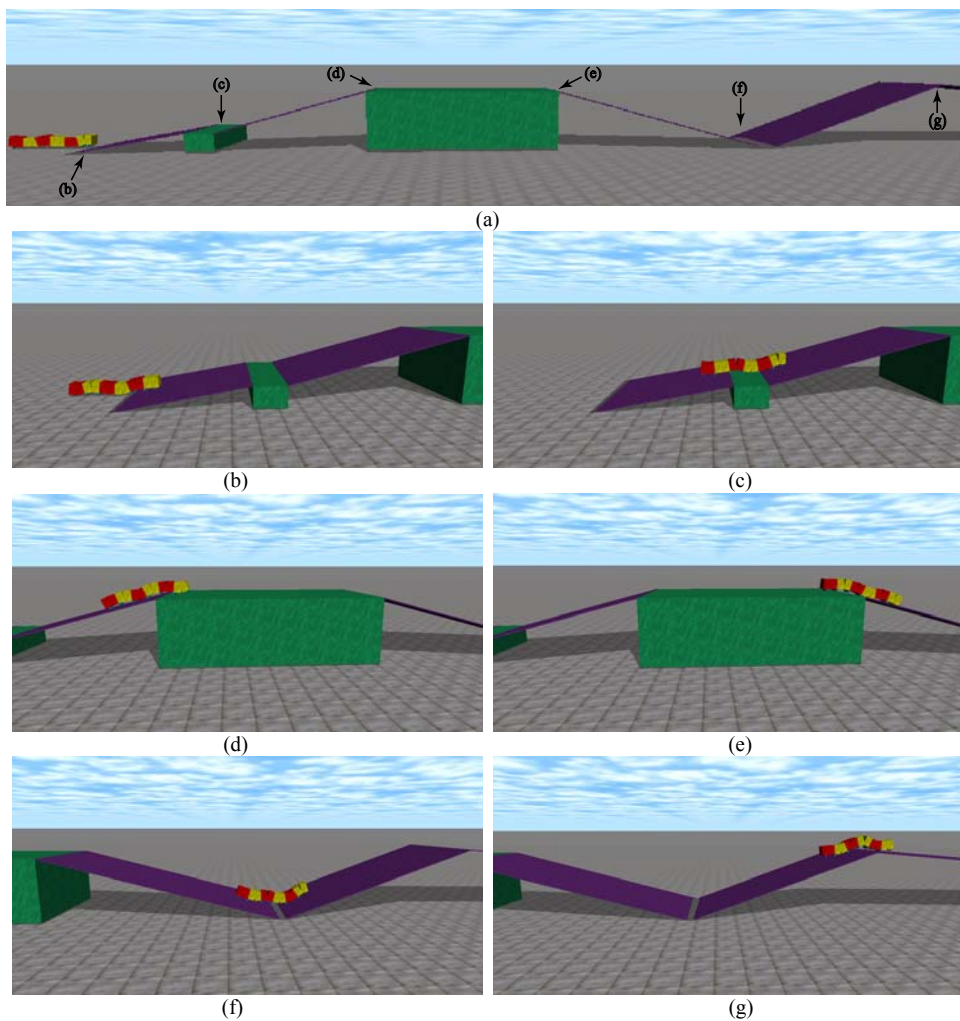

Figure 8. Simulation for adaptive locomotion. (a) The overall of the constructed environment. Slopes with different angles (from 5° to 20°) and boxes are spliced together. (b)-(g) Adaptation of caterpillar-like locomotion. The robot managed to climb over these slopes with the help of sensory feedback.

(3) For each joint i, check which case it belongs to and determine reaction time constant δ as well as the amount of afferent stimulus λ;

(4) Calculate the output of the sensory neuron in each oscillator;

(5) Calculate the output of all the oscillators;

(6) Apply the result to all the joints.

Figure 8(b)-(g) illustrates the procedure of the adaptive locomotion for the robot climbing over the slopes. Note that the position of each sub-figure (b)-(g) corresponds to a position labelled in Figure 8(a). The feasibility of the control mechanism is validated via the simulated result.

5.2 On-site experiment

A real pitch-pitch connected modular robot with 6 aluminium modules was constructed for the on-site experiment. The aluminium module, driven by a RC servomotor (Futaba 3003), weighs 0.15kg and has a length of 72mm with a cross-section of 52×52mm. Each end of the module is surrounded by six touch switches (Omron D2MQ). In this experiment, only the touch switches on the bottom are used to collect the terrain information. The testing scenario built with boards is *6m* long and is the same size as the one constructed in the simulation. Also, the on-site experiment was examined with the same set of evolved parameters, as shown in Table 4.

Figure 9 shows a series of snapshots taken from a video recorded during the experiment[1]. The robot successfully used adaptive behaviours when climbing over the slopes in Figure 8(a). It was found that the body shape of the robot can be bent flexibly to adapt to the terrain with the help of integrated sensory information of the

---

1 The video of the on-site experiment is available at: http://tams-www.informatik.uni-hamburg.de/people/li/on-site_experiment.mp4.

environment. In contrast, the caterpillar-like robot without sensory feedback gets stuck at the second slope (the position as in Figure 9(b)).

Table 4. Evolved parameters.

Module	Parameters				
	λ_1	λ_2	λ_3	δ_m	δ_r
No. 1	0.40	-	-	6	1.5
No. 2	0.55	0.10	-0.25	17.5	1.5
No. 3	0.10	0.25	-0.30	11	3.5
No. 4	0.25	0.45	-0.35	8	9
No. 5	0.05	0.65	-0.15	6.5	17
No. 6	-	0.40	-	17	3.5

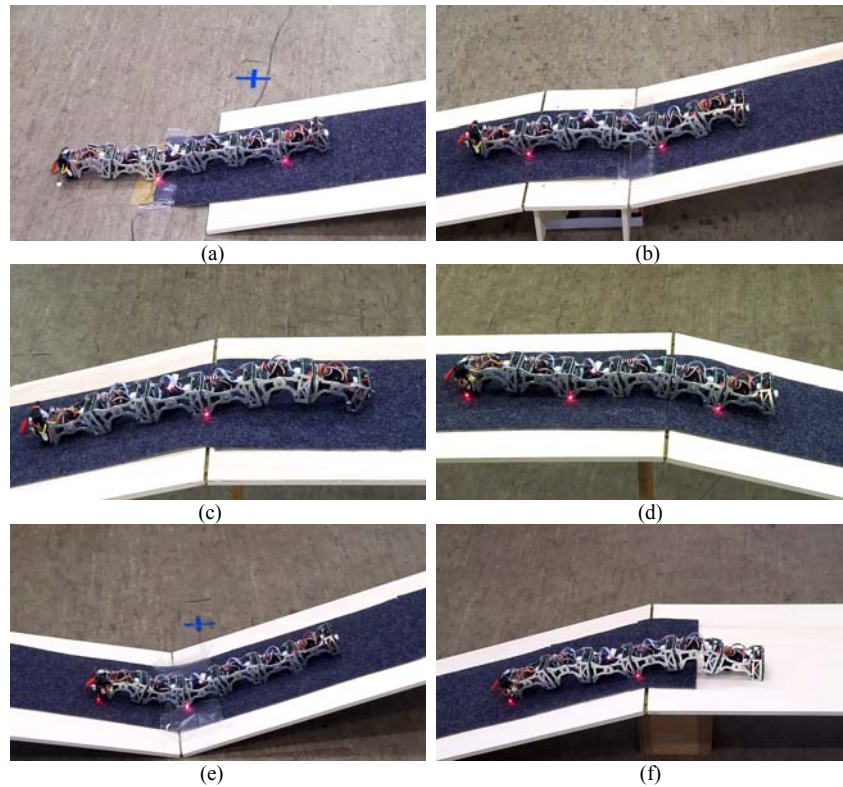(a)　　　　(b)

(c)　　　　(d)

(e)　　　　(f)

Figure 9. On-site experiment. (a)-(f) Scene of the robot climbing over slopes in the environment.

Figure 10 shows an example from the on-site experiment where a module in the centre of the robot (module 4) is tracked and analysed. In Figure 10, the three curves from the top to the bottom represent the afferent stimuli, sensory input and joint output, respectively. The vertical lines indicate the moments when the robot is climbing over slopes shown correspondingly in Figure 9(a)-(f). The output of the module is shifted when the robot is climbing over slopes. An exception is the last slope (see Figure 9(f)). The reason is that during the climbing of the last slope, all modules can contact the terrain periodically. This means no stimuli are passed to the modules and thus their corresponding joint oscillation would not be shifted. Based on the results, we conclude that the proposed control mechanism is effective in realising adaptive locomotion pattern for caterpillar-like robot.
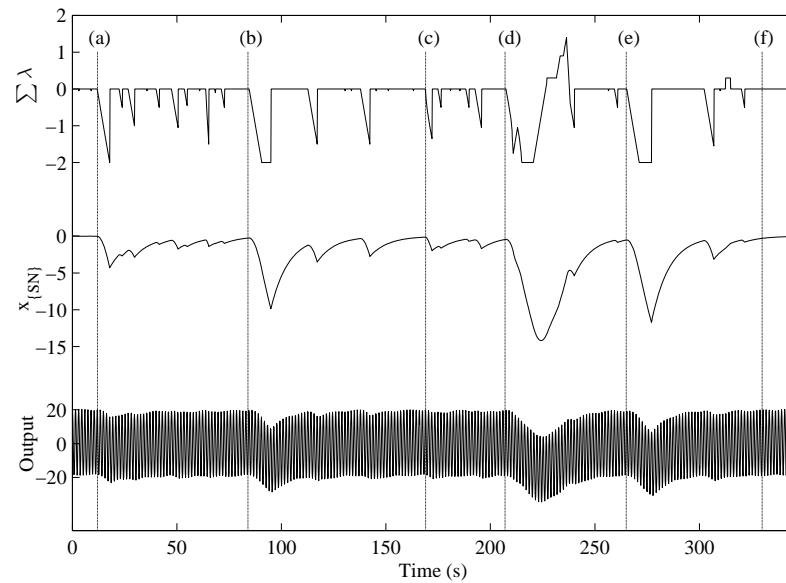

Figure 10. Tracked data of module 4 during the on-site experiment, namely the afferent stimuli over time (top); the sensory input generated by the corresponding sensory neuron (middle); and output (bottom).

6 Conclusion

In this paper, a framework of closed-loop control for adaptive locomotion of a caterpillar-like robot was presented. Biologically inspired by the spinal circuit of lampreys, sensory feedback is successfully integrated into a new CPG model by means of three components, namely a sensor processor, a reaction maker, and a parameter modulator that control the robot locomotion together. To achieve adaptive behaviours, reaction strategies are further optimised by a genetic algorithm. Both simulation and on-site experiment show the effectiveness of the approach in realising adaptive locomotion of a caterpillar-like robot.

For future work, two aspects need to be improved. First, for the reaction maker, more specific patterns should be identified so that more reaction behaviours can be made. Second, energy consumption should be considered during the robot locomotion, which would be helpful in making the locomotion not only faster and more stable, but also more efficient.

References

[1]   S. Grillner. Neurobiological bases of rhythmic motor acts in vertebrates. Science. 1985; 228: 143-149.

[2]   S. Aoi, K. Tsuchiya. Stability analysis of a simple walking model driven by an oscillator with a phase reset using sensory feedback. IEEE Transactions on Robotics. 2006; 22: 391- 397.

[3]   G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi and G. Cheng. Learning CPG-based biped locomotion with a policy gradient method: application to a humanoid robot. Int. J. Robot. Res. 2008; 27: 213–228.

[4] X. Wu, S. Ma. Adaptive creeping locomotion of a CPG-controlled snake-like robot to environment change. Auton. Robot. 2010; 28: 283-294.

[5] X. Wu, S. Ma. Sensor-driven neural controller for self-adaptive collision-free behavior of a snake-like robot. In proc. of IEEE Int. Conf. on Robotics and Automation (ICRA). Shanghai, 2011; 191-196.

[6] X. Wu, S. Ma. Neurally controlled steering for collision-free behavior of a snake robot. IEEE Trans. on Control Systems Technology. 2013; 1: 1-7.

[7] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz and H. Choset. Parameterized and scripted gaits for modular snake robots. Advanced Robotics. 2009; 23: 1131-1158.

[8] J. K. Ryu, N. Y. Chong, B. J. You and H. I. Christensen. Locomotion of snake-like robots using adaptive neural oscillators. Intell. Serv. Robot. 2010; 3: 1-10.

[9] L. Zhu, Z. Chen and T. Iwasaki. Oscillation, orientation, and locomotion of underactuated multilink mechanical systems. IEEE Trans. on Control Systems Technology. 2012; 1: 1-12.

[10] G. Yang, S. Ma, B. Li and M. Wang. A hierarchical connectionist CPG controller for controlling the snake-like robot's 3-dimensional gaits. In proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS). Portugal, 2012; 822-827.

[11] T. Sato, T. Kano, A. Ishiguro. A snake-like robot driven by a decentralized control that enables both phasic and tonic control. In proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS). San Francisco, 2011; 1881-1886.

[12] T. Sato, W. Watanabe and A. Ishiguro. An Adaptive decentralized control of a serpentine robot based on the discrepancy between body, brain and environment. In proc. of IEEE Int. Conf. on Robotics and Automation (ICRA). Alaska, 2010; 709-714.

[13] N. M. Norzalilah, S. Ma. A simplified CPGs network with phase oscillator model for locomotion control of snake-like robot. In proc. of IEEE Int. Conf. on Robotics and Biomimetics (ROBIO). Guangzhou, 20012; 1299-1304.

[14] C. Liu, Q. Chen and D. Wang. CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots. IEEE Trans. Systems, Man and Cybernetics, Part B. 2011; 41: 867–880.

[15] H. Kimura, Y. Fukuoka, A. H. Cohen. Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. Int. J. Robot. Res. 2007; 26: 475-490.

[16] L. Righetti, A. J. Ijspeert. Pattern generators with sensory feedback for the control of quadruped locomotion. In proc. of IEEE Int. Conf. on Robotics and Automation (ICRA). Pasadena, 2008; 819-824.

[17] X. Zhang, H. Zheng and L. Chen. Gait transition for a quadrupedal robot by replacing the gait matrix of a central pattern generator model. Advanced Robotics. 2006; 20: 849-866.

[18] W. Zhao, Y. Hu, L. Zhang and L. Wang. Design and CPG-based control of biomimetic robot fish. IET Control Theory Application. 2009; 3: 281-293.

[19] A.J. Ijspeert. Central Pattern Generators for Locomotion Control in Animals and Robotics: a review. Neural Networks. 2008; 21: 642-653.

[20] Ö. Ekeberg. A combined neuronal and mechanical model of fish swimming. Biol. Cybern. 1993, 69: 363-374.

[21] K. Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biol. Cybern. 1985, 52: 367-376.

[22] A. J. Ijspeert and A. Crespi. Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In proc. of IEEE Int. Conf. on Robotics and Automation (ICRA). Roma, 2007; 262-268.

[23] S.L. Hooper. Central pattern generators. Current Biology. 2000; 10: 176-177.

[24] S. Mezoff, N. Papastathis, A. Takesianm, B.A. Trimmer. The biomechanical and neural control of hydrostatic limb movements in manduca sexta. J. Exp. Biol. 2004; 207: 3043-3054.

[25] H. Zhang, W. Wang, J. Gonzalez-Gomez, J. Zhang. Design and realization of a novel modular climbing caterpillar using low-frequency vibrating passive suckers, Advanced Robotics. 2009; 23: 889-906.

[26] G. Matthews. NEUROBIOLOGY, Molecules, Cells and Systems, second edition. Blackwell Science; 2001.

[27] G. Li. Hierarchical control of limbless locomotion using bio-inspired CPG model. Doctoral dissertation, University of Hamburg; 2013.

[28] H. Zhang, J. Gonzalez-Gomez, J. Zhang. A new application of modular robots on analysis of caterpillar-like locomotion. In proc. of IEEE Int. Conf. on Mechatronics. Malaga, Spain, 2009; 1-6.

[29] Open dynamics engine. Available from: http://www.ode.org/.

[30] GAlib. Available from: http://lancet.mit.edu/ga/.
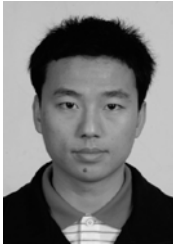
Guoyuan Li received both his Bachelor (2006) and Master (2009) of Engineering from the Department of Computer Science of Chongqing University, Chongqing, China. He is currently working toward a Ph.D. degree at the Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, University of Hamburg, Germany. His research interests include bio-inspired locomotion control and modular robotic design.

H. X. Zhang (M'04–SM'12) received Ph.D. degree in Mechanical and Electronic Engineering from Beijing University of Aeronautics and Astronautics, China, in 2003. From 2004, he worked as Postdoctoral Fellow at the Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, University of Hamburg, Germany. In February 2011, he finished the Habilitation at University of Hamburg. Dr. Zhang joined the Faculty of Maritime Technology and Operations, Aalesund University College, Norway in April 2011 where he is a full Professor on Robotics and Cybernetics. Currently, he also has a gift professorship on product and system design from the industry. The focus of his research lies on mobile robotics, especially on climbing robots and urban search and rescue robots, modular robotics, and nonlinear control algorithms. In these areas, he has published over 70 journal and conference papers and book chapters as author and co-author. Recently, he has received the best paper award at the IEEE/AEME AIM2008 conference, and three finalist awards for best conference paper at IEEE Robotics and Automation conferences.

J. W. Zhang (M'92) received both his Bachelor of Engineering (1986, with distinction) and Master of Engineering (1989) from the Department of Computer Science of Tsinghua University, Beijing, China, and his PhD (1994) at the Institute of Real-Time Computer Systems and Robotics, Department of Computer Science, University of Karlsruhe, Germany. Dr. Jianwei Zhang is full professor and head of TAMS, Department of Informatics, University of Hamburg, Germany.

Robin T. Bye (M'05) received his BE (2003, Honours Class 1), MEngSc (2004, in Systems and Control), and PhD (2009), all in Electrical Engineering, from the University of New South Wales, Sydney, Australia. In 2008 he began working at Aalesund University College, Aalesund, Norway, first as a Research Fellow and since 2010 as an Associate Professor in Automation Engineering. Dr. Bye's main research interest is intelligent control engineering, including diverse topics such as dynamic resource allocation, optimisation, computational intelligence, cybernetics, and human movement control systems.