Robotics and Autonomous Systems

AmphiBot I: an amphibious snake-like robot

Alessandro Crespi*, André Badertscher, André Guignard, Auke Jan Ijspeert

*School of Computer and Communication Sciences, EPFL, Swiss Federal Institute of Technology,
Lausanne, Station 14, CH-1015 Lausanne, Switzerland*

Abstract

This article presents a project that aims at constructing a biologically inspired amphibious snake-like robot. The robot is designed to be capable of anguilliform swimming like sea-snakes and lampreys in water and lateral undulatory locomotion like a snake on ground. Both the structure and the controller of the robot are inspired by elongate vertebrates. In particular, the locomotion of the robot is controlled by a central pattern generator (a system of coupled oscillators) that produces travelling waves of oscillations as limit cycle behavior. We present the design considerations behind the robot and its controller. Experiments are carried out to identify the types of travelling waves that optimize speed during lateral undulatory locomotion on ground. In particular, the optimal frequency, amplitude and wavelength are thus identified when the robot is crawling on a particular surface.
© 2004 Elsevier B.V. All rights reserved.

Keywords: AmphiBot I; Amphibious snake robot; Central pattern generator; Oscillator; Crawling

1. Introduction

This project aims at constructing a biologically inspired amphibious snake-like robot, called AmphiBot I. The goals of the project are two-fold: (1) to take inspiration from snakes and elongate fishes such as lampreys to produce a novel type of robot with dexterous locomotion abilities, and (2) to use the robot to investigate hypotheses of how central nervous systems implement these abilities in animals.

The project does not aim at mimicking a snake or a lamprey per se, but to take inspiration of their body shape and their neuronal control mechanisms to develop novel types of robots that exhibit dexterous locomotion. Snake-like robots are indeed among the most flexible and versatile mobile robots. In particular, their long but thin body and its division in several small segments make them well-suited to a large number of applications. Such applications include, for example, exploration and inspection tasks (e.g. in areas that are inaccessible to humans, such as pipes) and the participation to *search and rescue* missions (e.g. in a collapsed building or a flooded zone).

While a variety of different snake-like robots have been constructed (see Section 2), the main features of

* Corresponding author. Tel.: +41 21 693 66 30;
fax: +41 21 693 37 05.

E-mail addresses: alessandro.crespi@epfl.ch (A. Crespi);
auke.ijspeert@epfl.ch (A.J. Ijspeert).

our robot are (1) to be amphibious and capable of both swimming and lateral undulatory locomotion, and (2) to be controlled by a controller that is inspired by *central pattern generators* found in vertebrate spinal cords.

In the next sections, we will first make a short overview of the biological background of the project (i.e. the locomotion of snakes) and of related works. We will then describe the design considerations underlying our project, followed by a detailed description of the hardware and software of the robot. Experiments are carried out to identify the types of travelling waves that optimize speed during lateral undulatory locomotion on ground. We finish the article with a description of future work and a short conclusion. This article is an extended version of a paper published elsewhere [1].

2. Snake locomotion

Four main different locomotion modes have been documented [2–4] in snakes: lateral undulation (also called serpentine locomotion), concertina, sidewinding and rectilinear. Several other gaits exist, however, they are used only by a restricted number of snake species in somewhat special situations (tree climbing, jumping, etc.). Sometimes, depending on the environment, snakes use more than one locomotion mode at the same time, having a locomotion mode for one part of the body and another one for the other part ([5] as cited by [3]).

The *lateral undulatory* mode, characterized by a lateral S-shaped wave travelling from head to tail, is the most common and efficient one, and almost all snakes use it. Swimming snakes move the body practically in the same way [6]. This type of swimming is called *anguilliform swimming* among elongate fishes, such as eels and lampreys. In the *concertina* mode, part of the snake's body is pushed against a surface forming a small number of waves: by moving these waves, and the corresponding contact points, the snake progresses. This mode is generally used when the snake has to move along a straight path or when the friction coefficients of the floor do not allow lateral undulatory locomotion; however this is a rather inefficient mode and is seldomly used only when needed. *Sidewinding* is used by desert snakes that need to move on sand; in this mode, the snake lifts a part of the body to maintain only a few contact points with the ground, using them to move the rest of the body. The *rectilinear* mode is obtained

by cyclically "fixing" parts of the skin to the ground using scales, and then moving the backbone forward with respect to the skin, and finally releasing the scales allowing the skin to move forward. This locomotion mode is generally used only by big snakes (like boas), because their weight makes the lateral undulation inefficient. As its name says, rectilinear locomotion does not produce lateral undulations like the other ones.

Our robot will use lateral undulatory locomotion. To achieve this type of locomotion, an issue is of fundamental importance: the friction coefficients between the snake and the ground have to be directional. For each segment of the body (a snake has as many segments as the number of vertebrae—between 100 and 400 depending on the species), there must be a low friction coefficient in the tangential direction (the direction in which the segment is moving) and a high friction coefficient in the perpendicular direction, in order to avoid lateral displacement of the segment. This directional friction is obtained in snakes by the particular structure of the skin. A similar mechanism is used when swimming: due to the elongate shape, propulsion is produced by the combination of a low drag coefficient in the tangential direction and a higher one in the perpendicular directions.

2.1. Central pattern generators (CPGs)

Locomotion in vertebrates is controlled by central pattern generators, which are networks of neurons that can produce coordinated oscillatory signals without oscillatory inputs [7]. In vertebrates, CPGs for locomotion are located in the spinal cord and distributed in multiple oscillatory centers.

A typical example of CPG for anguilliform swimming is found in the lamprey. The lamprey is one of the earliest and simplest vertebrates. It has no paired fins and swims by propagating an undulation along its body, from head to tail. Its CPG has been extensively studied [8–11]. It is composed of 100 segmental networks, with each segmental network containing at least two oscillatory centers, one for each side of the spinal cord (left and right). When the isolated spinal cord is placed in an excitatory bath, it starts to produce an oscillatory neural activity called *fictive swimming* that is very similar to that observed during intact locomotion. The CPG will then produce oscillations with a phase lag between neighboring segments such that a travelling wave

is propagated from head to tail. When the stimulation of the network is increased (higher concentration of the excitatory bath), the frequency of oscillation increases, which is associated with an increase of the speed of swimming.

CPGs are an interesting source of inspiration for controlling robots: (1) they implement a control scheme that can be implemented in a distributed fashion; (2) they require only simple command signals to produce complex coordinated multi-dimensional output signals; and (3) they easily incorporate sensory feedback and take mechanical perturbations into account.

3. Currently existing snake and lamprey robots

Snake robots can be classified into two main groups:

- robots that move using powered wheels (i.e. a torque is applied on the axis of the wheels, which are in contact with the ground, producing a rotation and consequently a movement);
- robots that move by applying torques on the joints between the segments. Among these robots, some have passive wheels.

Robots using powered wheels are simpler to control: the design techniques are well known and standard algorithms for the control of mobile robots can be used; however, the resulting locomotion is completely artificial and the wheels may not be adequate in every environment. Robots of this type are often developed for inspection tasks in difficultly accessible zones [12,13] and are currently used, for example, for the inspection of pipes [14]. On the other side, robots that use powered joints instead of powered wheels are more complicated to design, and the control algorithms that can be used are partially unexplored. As we aim to design a biologically inspired snake robot that can both crawl and swim with powered joints, we are mainly interested in this second approach.

One of the first known snake robots was built by Hirose and co-workers at the end of 1972 [15]. He generically named this kind of robot an *active cord mechanism* (ACM). After this first prototype, he built some other snake robots [16]. A huge snake robot has been developed in 1992 at Caltech [17]. The Jet Propulsion Laboratory of the NASA presented in 1994

a serpentine robot [18]. Miller developed several prototypes of snake robots; among them the last one, S5 [19], has a very realistic lateral undulatory gait (its locomotion is probably the most similar to a biological snake, compared to other snake robots). Saito and co-workers presented in 2002 a simple snake robot used to validate some theoretical results [20]. Conradt and Varshavskaya [21] developed WormBot, a snake-like robot controlled by local CPGs. For a more detailed review of snake robots, see [4,22].

Swimming snake robots (also referred to as lamprey robots or eel robots) are rarer. They are generally designed to imitate the anguilliform swimming of the eel (or the very similar one of the lamprey). Several theoretical papers have been written on this subject, but there are only a few real robotic realizations. The robots in this category that are the most interesting are the eel robot *REEL II* [23] and the lamprey robot built at Northeastern University [24]. In principle, these eel and lamprey robots could be adapted to terrestrial locomotion, but such experiments have not been reported. To the best of our knowledge, there is currently only one amphibious snake-like robot, the HELIX-I ([25] as cited by [26]; [27]), that can both swim in water and crawl on the ground (although ground locomotion is not described in the papers).

3.1. Control methods

Two broad classes of control methods have been used with snake robots. The first class can be described as *trajectory-tracking control*. It uses predefined gait patterns, usually computed as sine waves, that are tracked with a feedback controller (e.g. a PID controller). Typically, the control is *open-loop*: the setpoints of the joints are calculated and sent to the motor controllers without any form of feedback (the only feedback present in the system is the one used by the PID controller). Examples of this approach include [16,28].

The other class can be described as *online gait generation control*. In this case, gaits are not predefined in advance, but generated online during locomotion. These approaches can, therefore, better deal with perturbations and irregular terrains. Most of these approaches are model-based, i.e. they rely on a kinematic or dynamic model of the robot's locomotion in order to design control laws for the gait generation.

Examples of control based on kinematic models include [29,30]. Examples of control based on dynamic models include [31–33]. Among these, the approach by Ono and co-workers [33] is interesting in that they use a self-excitation principle to generate gaits that are close to the natural vibration mode of the robot.

We will here explore another way of doing online gait generation. Our approach is CPG-based (see Section 2.1) and uses a system of coupled nonlinear oscillators to generate gaits online. Feedback terms can be included in the dynamical system to allow smooth online modifications of the gaits. Such an approach does not need an explicit model of the robot. The model is only implicit, and the CPG can be considered as a particular feed-forward controller. The interesting aspect of this approach is that gaits smoothly adapt to perturbations and modifications of the control parameters (e.g. the frequency and the amplitude of the travelling waves). A similar approach has been presented in [21]. The main difference between this approach and ours is that we use coupled nonlinear oscillators instead of phase oscillators. Unlike phase oscillators (which have no explicit amplitude state variable), nonlinear oscillators have the interesting property to have amplitude state variables that exhibit limit cycle behavior and that can directly be used to command the joints.

4. Design considerations

Our amphibious snake-like robot, AmphiBot I, is designed to present the following characteristics:

- To be modular. We aim at having a robot that is composed of multiple identical elements. This allows us to quickly adjust the length of the robot by adding or removing elements, as well as to replace defective elements.
- To have distributed actuation, power and control. In order to be truly modular, each element carries its own DC motor, battery, and microcontroller.
- To be waterproof. Each individual element is made waterproof (as opposed to having a coating covering a chain of elements). This facilitates modularity and ensures that a leakage will only damage a single element.
- To be slightly buoyant. We aim at having a robot that passively returns to the surface of the water when in-

active. Furthermore, we construct the elements such that the center of gravity is placed below the geometrical center, in order to obtain a vertical orientation that self-stabilizes in water.
- To have large lateral surfaces for good swimming efficiency.
- To have asymmetric friction for the lateral undulatory locomotion (lower friction coefficient in the longitudinal axis compared to the perpendicular axis).
- To be controlled by a CPG composed of coupled non-linear oscillators.
- In its current form, to be remotely controlled in terms of speed and direction commands, but otherwise have an onboard locomotion controller for coordinating its multiple degrees of freedom.

In the current version of the robot, all these desired characteristics have been implemented (see next sections), except for the buoyancy, and the onboard locomotion controller.

5. Hardware

AmphiBot I is modular and constructed out of several identical segments, named *elements* (Fig. 1). In the current prototype, each element has a single degree of freedom, and elements are fixed such that all axes of rotation are aligned. Each element consists of four structural parts: a body, two covers and a connection piece. All parts are molded using polyurethane. The Li-Ion battery is directly incorporated into the bottom cover when the polyurethane is cast in the mould. To ensure the waterproofing of the robot, O-rings are placed between each cover and the body, and around the output axis. An element has a length of 7 cm and a section of 5.5 cm by 3.3 cm. The current robot has seven actuated elements. Asymmetric friction is created by passive wheels or skates, which can be easily fixed to the elements using adhesive velcro stripes. Here all results are obtained with wheels.

In each element, there are two printed circuits (one for the power supply/battery charger and one for the motor controller), a DC motor and a set of gears. Two different voltages are used inside an element: 3.6 V and 5 V. The first one is the typical value of a Li-Ion battery and is only used to power the motor; the second
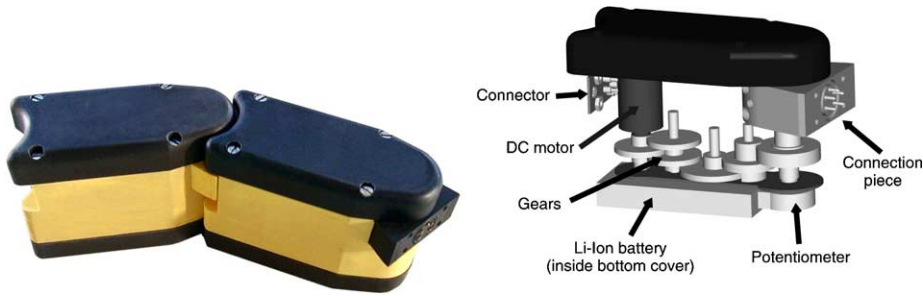
Fig. 1. Left: two connected elements; right: internal structure of an element.

one is used to power the electronics. When the robot is battery-powered (no external power source is connected), the motor is directly powered using the battery, without any intermediary regulator or converter, and the 5 V used by the electronics are generated with a capacitive charge-pump step-up converter (LTC 3200). When an external (5 V) power source is connected, the 3.6 V for the motor is generated using a low-efficiency diode to create a voltage drop, and the electronics are directly powered using the external source. When the external power source is present, the battery could also be charged if this is necessary; for this reason a small battery charger (LTC 1733) is part of the power supply circuit. The charger can be enabled or disabled by the microcontroller, using an *enable* signal. The battery has a capacity of 600 mAh, which is enough to power the element for an average time of approximately 2 hours of continuous use (but this largely depends on the movements that the robot has to do and on the external constraints applied to it). An empty battery can be charged in approximately 1 hour.

The motor controller (Fig. 2) is built with a PIC microcontroller (PIC 16F876) and some external components. The motor has a magnetic encoder, which generates 16 impulsions for every complete rotation of the axis. This encoder is connected to a LS 7084 quadrature detector that filters and decodes the signals of the magnetic encoder, generating a clock signal and a direction flag; these two signals are sent to the microcontroller, allowing it to track the current position of the motor. A 10 kΩ potentiometer is fixed to the output axis (after the reduction gears) and is connected to an analog input of the PIC; this potentiometer can be used to read the absolute position of the axis (for example, when the robot is switched on, or to detect possible skews between the

position measured with the magnetic encoder and the real one).

The motor coil is powered through a SI 9986 H-bridge, which supports currents up to 1 A. The H-bridge is driven by the microcontroller using a Pulse-Width Modulation (PWM) signal, allowing the speed of the motor to be changed.

Between the H-bridge and the motor, a $0.2\,\Omega$ resistor causes a voltage drop. The resistor is connected to the input of an INA 146 operational amplifier, the output of which is connected to one of the analog inputs of the microcontroller, therefore allowing a measure of the current used by the motor, and then indirectly of its torque. The negative voltage (-5 V) required to power the operational amplifier is obtained using a small capacitive inverter regulator (MAX 1719).

The 0.75 W DC motor (having a maximum torque of 1.2 mN m) drives a set of reduction gears with a reduction factor of 400, and an efficiency around 60%. The output axis of the gears is fixed to the aforementioned potentiometer and to the connection piece fixed to the next element. Considering the typical working speed of the motor and the reduction of the gears, a maximum oscillation frequency of approximately 0.3 Hz can be obtained if the full amplitude ($\pm 45°$) is used.
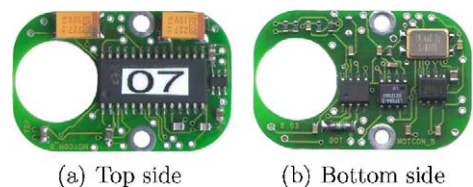

Fig. 2. The motor controller circuit (1:1 scale).
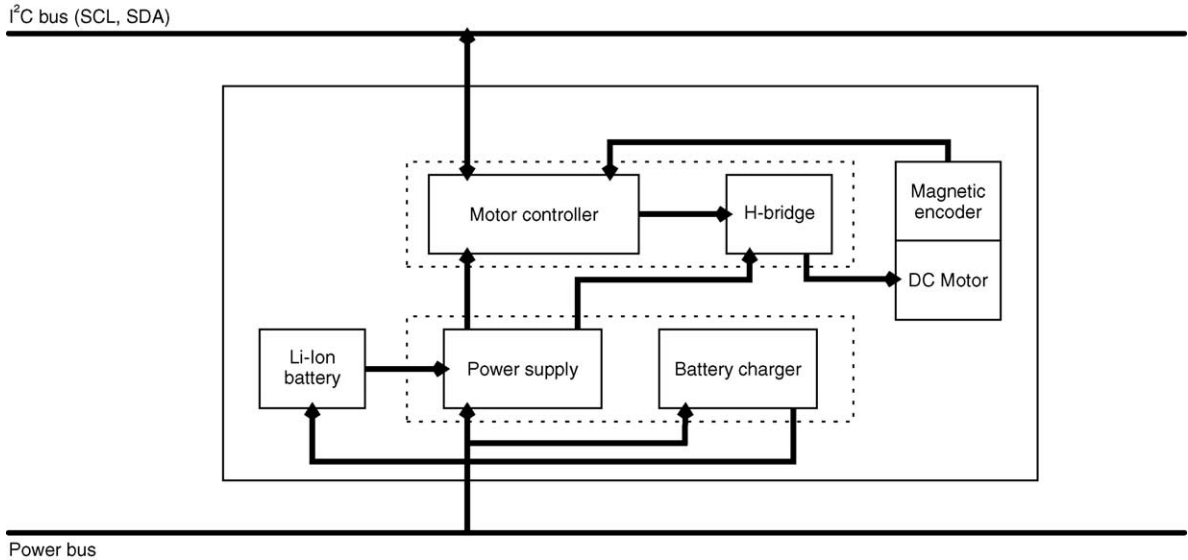
I²C bus (SCL, SDA)

Fig. 3. Schematic of the main blocks inside an element.

Five wires, passing through the (internally empty) axis, are connected to the contacts that are molded into the connection piece; four of them are used to pass the I²C bus and the external power source all along the robot. The fifth wire is currently unused and is reserved for future applications The structure of the electronics of an element is shown in Fig. 3. The complete robot with passive wheels is shown in Fig. 4.

The head of the robot is empty: being the first element, it neither needs a motor nor any controller or power supply. The last element (tail) is identical to the others; a special connection piece is currently fixed to it, allowing the bus and the power line to be connected to external equipments (recharging station, PC interface). Wireless communication capabilities will be added in a near future in order to allow the use of the robot in a tether-less mode.

6. Control

The control of locomotion of the robot is based on a system of coupled nonlinear oscillators that mimic central pattern generators found in vertebrates (see also Section 2.1). In previous work, we have modelled CPGs for swimming and walking using neural network simulations [34,35]. Here we instead use nonlinear oscillators as building blocks for constructing CPGs. Neural networks and coupled nonlinear oscillators exhibit very similar limit cycle behaviors [36]. The use of nonlinear oscillators instead of neural network oscillators allows us to reduce the number of state variables and parameters in the models, and, therefore, to develop controllers that are better suited to be implemented in a distributed fashion on the modular robot.[1]

We use the following nonlinear oscillator:

$$
\begin{cases}
\tau \dot{v} = -\alpha \dfrac{x^2 + v^2 - E}{E} v - x \\
\tau \dot{x} = v
\end{cases}
\tag{1}
$$

In this equation, x and v are state variables, and E, τ and α are positive constants that control the behavior of the oscillator. In our implementation, the variable x will determine the desired angle of the corresponding robotic element.

This oscillator has the interesting property that its limit cycle behavior is a sinusoidal signal with am-

---

[1] Nonlinear oscillators are also more suitable than neural networks for the implementation using small microcontrollers, which have very small amounts of memory and limited computing speeds; neural networks would require more memory (particularly for the higher number of parameters) and a more powerful processor, which would require more energy, thus reducing the battery life. Moreover, building a really distributed neural network is difficult, due to the high amount of data that would have to be transferred on the bus.

Fig. 4. The robot (here with passive wheels) in the experimental setup.

plitude $\sqrt{E}$ and period $2\pi\tau$. x indeed converges to $\tilde{x}(t) = \sqrt{E} \sin(t/\tau + \phi)$, where ϕ depends on the initial conditions (Fig. 5), from any initial conditions (except $(x_i, v_i) = (0, 0)$ for all i, which is an unstable fixed point). The E (*energy*) parameter, therefore, controls the amplitude of the oscillator's limit cycle, and the τ parameter controls its period. This kind of equation can be numerically integrated using simple Euler or Runge–Kutta methods. These methods can be adapted to be used on microcontrollers without particular problems.
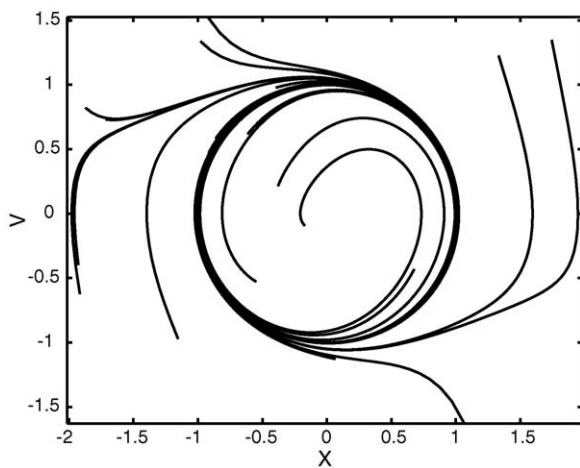

Fig. 5. Limit cycle behavior; time evolution of the nonlinear oscillator with different random initial conditions.

A specific gait pattern will be obtained by coupling several oscillators together, in our case one oscillator per element. Couplings are created by projecting signals proportional to x and v states from one oscillator to the other:

$$
\begin{cases}
\tau\dot{v}_i = -\alpha \dfrac{x_i^2 + v_i^2 - E_i}{E_i} v_i - x_i + \sum_j (a_{ij}x_j + b_{ij}v_j) \\
\tau\dot{x}_i = v_i
\end{cases}
$$
(2)

The a_{ij} and b_{ij} constants define the coupling between the different oscillators (i.e. the influence that the jth oscillator has on the ith one).

The CPG used in this project is composed of a chain of oscillators (Fig. 6, left). For simplicity, we assume that only nearest neighbor connections exist between oscillators. We also assume that all oscillators are identical along the chain (except for the oscillators at the extremities, which do not receive signals from their missing neighbors). The coupling in chain is therefore defined by four parameters, $a_{i,i-1}, b_{i,i-1}$ for the rostral couplings and $a_{i,i+1}, b_{i,i+1}$ for the caudal ones (rostral means toward the head, caudal means toward the tail). By exploring the four-dimensional parameter space of different possible coupling weights between oscillators, it is easy to find couplings that produce stable travelling waves from head to tail necessary for lateral undulatory locomotion and anguilliform swimming (see Section 7).

We use a PD controller to compute the torques necessary to produce the desired angles x_i for the element i. The PD controller software contained in the PIC microcontroller is a DC motor controller, developed at the Autonomous Systems Laboratory, another laboratory of the EPFL. This program, completely written in assembler, allows the motor to be controlled in several ways (position control, speed control, torque control and some variants). The only control mode we consider here is the position control (based on a standard PD controller), because we need to control the angle between each couple of elements in order to generate the required gait patterns. The gait patterns are thus generated by constantly modifying the *setpoint* (desired position) of each element. In the current implementation, the PD controllers are onboard, while the gait patterns are generated by an offboard computer.
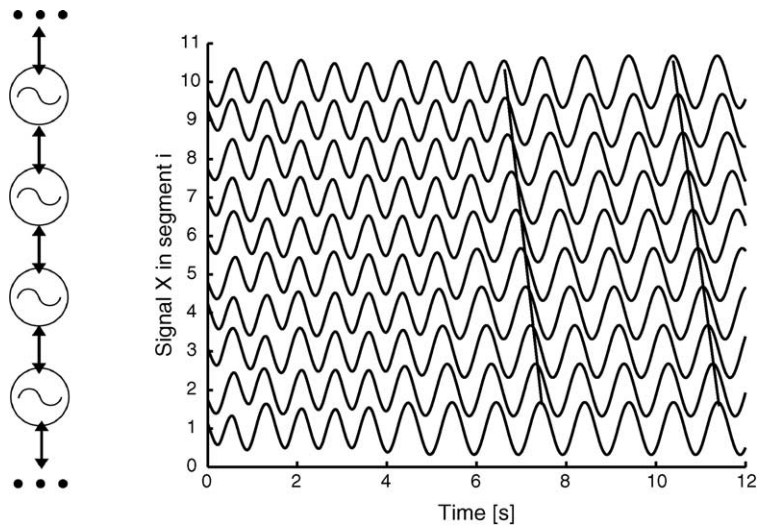
Fig. 6. Left: configuration of the body CPG; right: oscillations in a 10-oscillator chain. The oblique lines show that a travelling wave with a wavelength of approximately the length of the chain is obtained.

7. Results

7.1. Locomotion control with a CPG

Both the anguilliform swimming and the serpentine gaits require a travelling wave to be propagated from head to tail. After systematic exploration of the four-dimensional parameter space, we identified a set of solutions that spontaneously propagate a travelling wave from head to tail. Fig. 6 (right) illustrates the travelling waves generated by one particular solution (with $a_{i,i-1} = -0.9$, $b_{i,i-1} = 1.0$, $a_{i,i+1} = 0.0$, and $b_{i,i+1} = 0.0$, where $i = 1$ corresponds to the head oscillator). This particular controller produces a wavelength that is approximately the length of the 10-oscillator chain. An interesting feature of the controller is that the system rapidly stabilizes in a travelling wave, and this from any initial conditions (except from the unstable fixed point mentioned before).

By varying the parameters τ_i and E_i of the oscillators, one can easily adjust the period and the amplitude of the oscillations, respectively. Fig. 7 shows two examples when these parameters are abruptly changed for all oscillators. Despite the abrupt changes, the oscillations in the chain smoothly adapt to the new period and new amplitude. These parameters offer therefore the possibility to easily and smoothly adjust the speed of locomotion depending on the conditions.

One of the main motivations for using nonlinear oscillators is their ability to cope with transient perturbations. When correctly coupled, a chain of oscillators produces a stable limit cycle behavior to which the system will evolve from any initial conditions (except from the unstable fixed point mentioned above) and after any type of transient perturbation. Fig. 8 illustrates this property. At a given time, random perturbations are applied to all state variables x_i. After a short transitory period the system quickly and smoothly returns to the original travelling wave (Fig. 9).

The locomotion controller is currently being tested both in a dynamic simulation, and with a seven-joints (eight elements) real robot, using passive wheels to produce the directional friction required. Using Webots Dynamics [37], a dynamic simulation of articulated rigid bodies developed by Cyberbotics, we developed a simulation tool of the robot that allows us to test controllers in a physics-based model of the robot. A description of the simulator and a detailed analysis of the swimming and crawling gaits in simulation can be found in [38].

7.2. Identification of efficient travelling waves

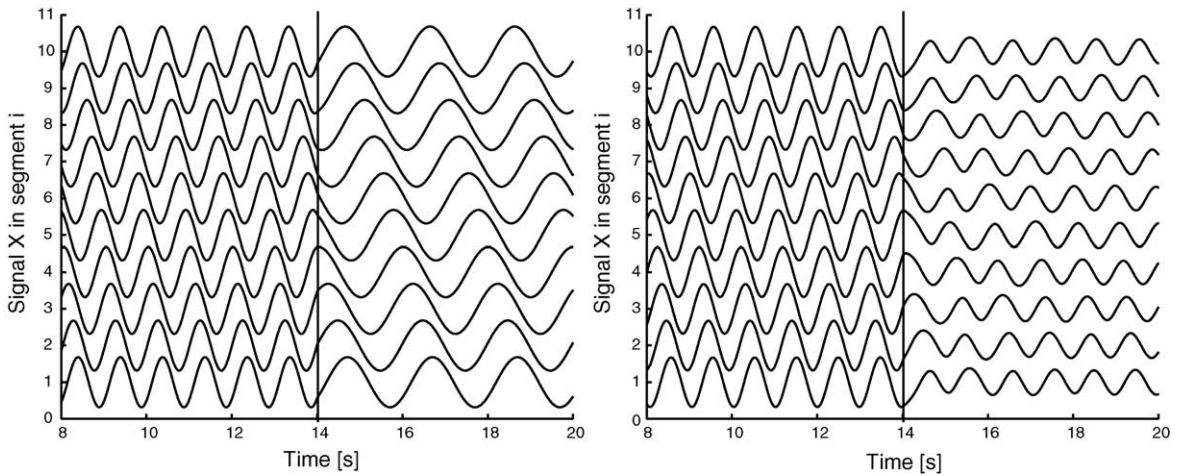To identify which types of travelling waves (in terms of wavelength and amplitude of oscillation) produce

Fig. 7. Left: modulation of the period by doubling the parameters τ_i at time $t = 14$ s; right: modulation of the amplitude by dividing the parameters E_i by a factor 4 at time $t = 14$ s.
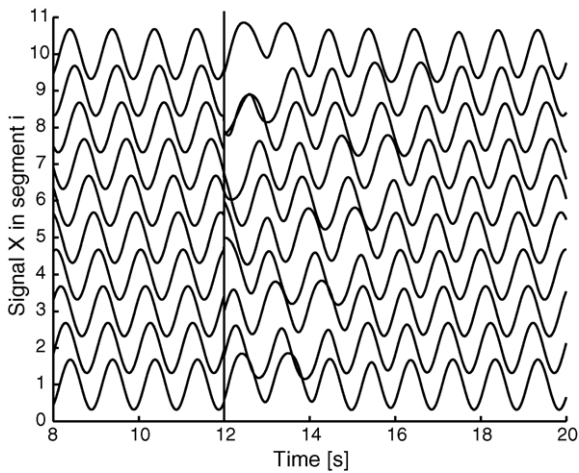

Fig. 8. Random perturbation of the state variables x_i at time $t = 12$ s.

the fastest locomotion gaits (using lateral undulation), for a given frequency of oscillation, we systematically tested the wave parameters in an acceptable range. Exceptionally, we used a simple open-loop sine controller to do this identification, instead of using CPGs, because the resulting waves are practically the same and the systematic change of the wavelength is simpler. The setpoint (desired angle) for the ith joint is generated as follows:

$$\theta_i = A \cdot \sin(2\pi \cdot \nu \cdot t + 2\pi \cdot \Delta\phi \cdot (i-1)) \qquad (3)$$

We performed a systematic search of different waves by varying the amplitude A, the phase lag $\Delta\phi$, and the frequency ν. We tested oscillation amplitudes A between $10°$ and $45°$, with steps of $5°$. The phase lag $\Delta\phi$ has been varied between $0.25/N$ and $1.5/N$, with a step of $0.25/N$ (where N is the number of actuated joints in the robot, i.e. currently 7). Note that a phase
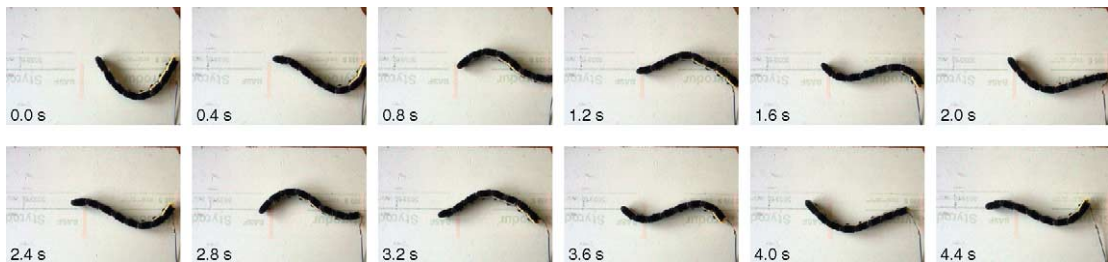

Fig. 9. The robot moving ($A = 30°$, $N \cdot \Delta\phi = 0.5$ and $\nu = 0.5$ Hz).

lag of $1/N$ corresponds to a wavelength of one body length, i.e. the body then makes one complete S-shaped wave. The oscillation frequencies were 0.25, 0.5 and 0.75 Hz.

The samples are sent in realtime from the controlling PC, through the I^2C interface, at the maximum possible communication rate (an average rate of 44 samples per second per joint has been measured). The speed of the robot has been measured by running the robot on a Styrodur® (rigid polystyrene foam) plain surface (this material has been chosen out of several different ones for its good properties of friction with the wheels of the robot) until the head reached the end of the surface, with a maximum run time of 60 s. The distance used to calculate the speed has been measured between the position of the robot's tail at $t = 0$ and the position of the tail at the end of the run. Runs have been executed only one time because no significative variance has been detected during preliminary experiments. The setpoints sent to the controllers and the real position of each joint have been recorded, allowing us to establish whether the desired joint trajectories are followed or not.

Depending on the parameters, the locomotion speed varied between 0 and 0.035 m/s (i.e. 0.06 body length per second). The maximum locomotion speed has been obtained with $v = 0.25$ Hz, $N \cdot \Delta\phi = 0.5$ and $A = 45°$. The results of the experiments are plotted
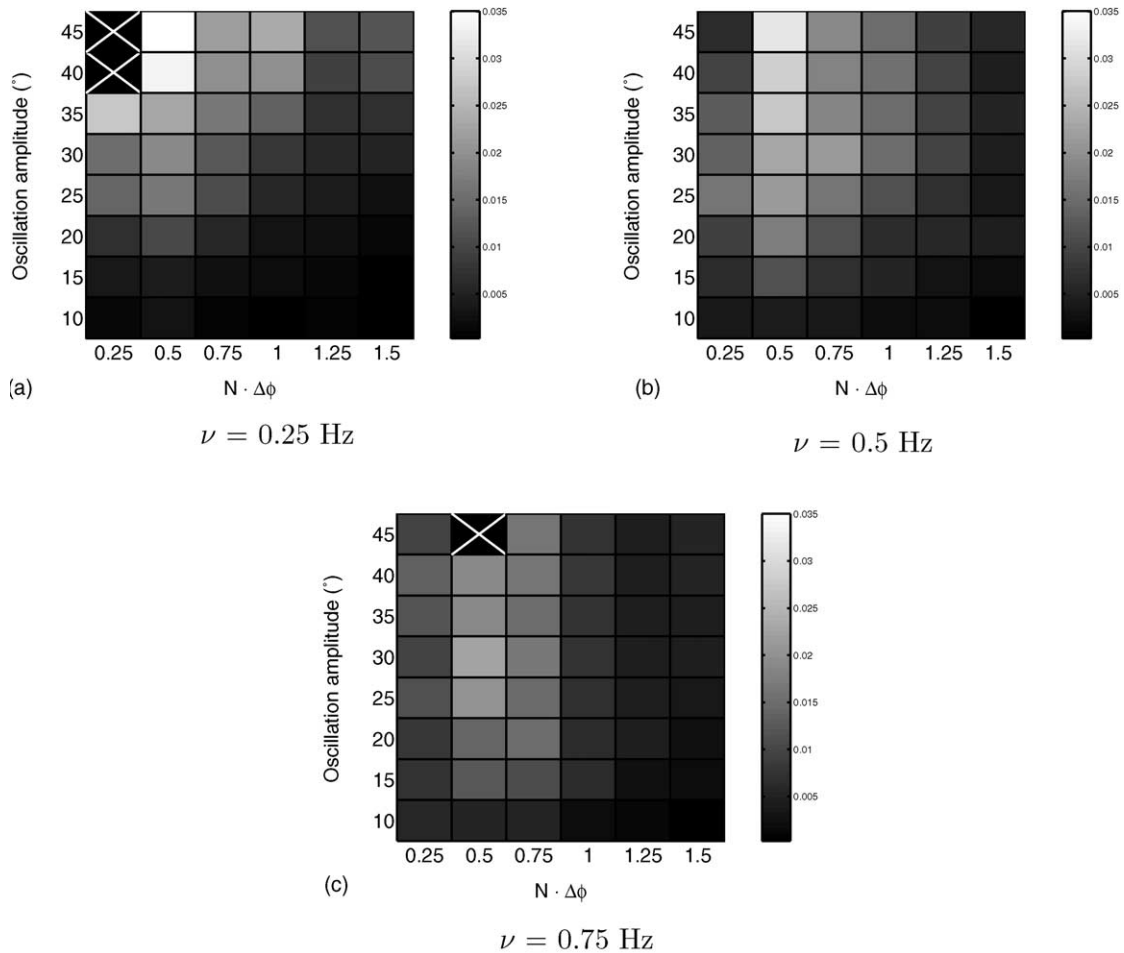

Fig. 10. Locomotion speeds (in m/s) with the different parameters.

in Fig. 10. It is possible to notice that only one optimum is present and that it is peaked (i.e. it is located in a small region of the parameter space); this means that it is important to find the right wave; otherwise, the locomotion is much slower.

It is easy to notice that there is a clear relationship between the oscillation amplitude and the locomotion speed: the higher the amplitude, the faster the locomotion. Amplitudes of $\pm 10°$ or less do not produce significant locomotion. The optimum is obtained with an oscillation amplitude of $\pm 45°$, which is the maximum amplitude that can be physically generated by the robot. A dependence on the phase difference is also easily visible, with an optimum around $N \cdot \Delta\phi = 0.5$ (i.e. with a wavelength equivalent to two times the length of the robot).

Results for some parameters (i.e. $N \cdot \Delta\phi = 0.25$, $\nu = 0.25$ Hz, $A > 40°$ and $N \cdot \Delta\phi = 0.5$, $\nu = 0.75$ Hz, $A = 45°$) are not available because the robot is completely unstable and falls on a side in very short times after the start of the experience. It is also important to note that the locomotion at $N \cdot \Delta\phi = 0.25$ is generally not dynamically stable (the robot falls on a side if the locomotion is instantly stopped) and is not a lateral undulatory locomotion, but looks like sidewinding. This is in part due to a small defect in the mechanical structure of the robot (its bottom is not completely straight but slightly curved along the chain), that needs to be corrected.[2] For the same reasons, the locomotion at $N \cdot \Delta\phi = 0.5$ is not perfectly rectilinear, but produces a slight lateral displacement (between 0.05 and 0.2 m, depending on the frequency).

The frequency also influences the locomotion speed, but in a less remarkable way compared to amplitude and phase difference; the optimum frequency is 0.5 Hz. At $\nu = 0.75$ Hz, the maximum speed of the motors does not allow the joints to reach the full oscillation amplitude of $\pm 45°$; the maximum amplitude that can be reached is about $\pm 20°$ therefore, the results with amplitudes greater than $\pm 20°$ are not really significant. The same is true for the larger amplitudes at $\nu = 0.5$ Hz, where the maximum oscillation amplitude is $\pm 30°$.

Comparing the results with those obtained in simulation [38], we can observe that the structure of the data is qualitatively similar, but the values are different. In particular, speeds of locomotion tend to be higher in simulation. These differences are mainly due to the higher number of active joints in the simulated robot and to the modelling of the contacts with the ground in the simulation, which are less "noisy" than those with the real robot (the fact that the bottom of the robot is not completely flat means that some elements tend to slide too much on the ground). Nevertheless, the simulation also demonstrates that an optimal phase lag exists for a given frequency and amplitude. The simulation shows that speeds of locomotion increase up to some optimal value ($A = 40°$) and tend to decrease for higher amplitudes. By improving the contacts with the ground, and to some extent increasing the range of motion of the joints, we therefore expect to be able to increase the speed of locomotion of the robot even more.

8. Future work

In addition to the current developments mentioned above, there is a large amount of work that can be done, mainly to enhance the current robot, and in particular:

- The waterproofness of the robot has been tested in water; the optimal parameters in this environment have to be determined. We expect swimming to be at least as fast as crawling, since the robot has been designed to have good swimming properties (large lateral surfaces).
- The robot should have the possibility to be completely autonomous. The current version can be independent from the energetic point of view, but not for the control; all control information is currently sent to the robot from an external source (i.e. a PC), using the I^2C bus. We plan to integrate a microcontroller or microprocessor based robot controller in a special element (for example, the head), in order to open the way to real autonomy.
- To demonstrate that nonlinear oscillators can be used for distributed control, we consider to implement a really distributed control running a nonlinear oscillator in each element's microcontroller. This will require some modifications to the actual *master–slave* bus, but should otherwise be fairly straightforward.

---

[2] We are now investigating the use of rubber connection pieces, which are not completely rigid (but still waterproof), in order to compensate the small mechanical inaccuracies of the robot and possible small asperities of the ground.

- It must be possible to control the robot using a sort of remote control; a (possibly bidirectional) wireless data link has thus to be realized. This may be fairly problematic as the water is a very bad medium for the propagation of electromagnetic waves. We are currently investigating which technology is best suited for underwater control.
- Requirements to achieve lateral undulatory locomotion on the ground are still to be analyzed in detail. As asymmetric friction is required for this type of locomotion, different ways to obtain it (e.g. skates or scales) are to be investigated.
- We currently have only one degree of freedom per element. This may be a problem in two cases: when the robot has to get over an obstacle (this would require some vertical flexibility) and if the robot falls on one side. In this last case the robot has still the possibility to successfully progress with caterpillar locomotion, but it is unable to rotate itself to recover the correct orientation.
- The current snake-like robot is a good base to build a salamander robot like those investigated in simulation in [35]. We are currently developing AmphiBot II, which will have special elements with simple limbs to add walking as an additional available gait.
- The gaits obtained with the robot have to be compared to biological data (i.e. to the gaits of real snakes of similar size), to highlight similarities and differences. This may lead to obtain useful data to enhance the robot.

9. Conclusions

This article presented the first developments in a project that aims at developing AmphiBot I, an amphibious snake-like robot capable of swimming and lateral undulatory locomotion. The design considerations behind the robot's hardware and software were presented. A CPG-based controller constructed out of a chain of coupled oscillators was implemented. The controller presents interesting features such as distributed control, robustness against perturbations, and ability to smoothly adapt the frequency and amplitude of oscillations, when control parameters are varied. The type of travelling waves (in terms of wavelength, amplitude of oscillation, and frequency) that produce the fastest locomotion gaits using lateral undulation with the robot have been identified.

References

[1] A. Crespi, A. Badertscher, A. Guignard, A.J. Ijspeert, An amphibious robot capable of snake and lamprey-like locomotion, in: ISR, 2004.

[2] J. Gray, The mechanism of locomotion in snakes, J. Exp. Biol. 23 (1946) 101–120.

[3] B.C. Jayne, Kinematics of terrestrial snake locomotion, Copeia 4 (1986) 915–927.

[4] K. Dowling, Limbless locomotion: learning to crawl with a snake robot, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1997.

[5] C. Gans, Biomechanics: An Approach to Vertebrate Biology, University of Michigan Press, Ann Arbor, Michigan, 1974.

[6] B.C. Jayne, Swimming in constricting (*Elaphe g. guttata*) and nonconstricting (*Nerodia fasciata pictiventris*) colubrid snakes, Copeia 1 (1985) 195–208.

[7] F. Delcomyn, Neural basis for rhythmic behaviour in animals, Science 210 (1980) 492–498.

[8] J.T. Buchanan, S. Grillner, Newly identified 'glutamate interneurons' and their role in locomotion in the lamprey spinal cord, Science 236 (1987) 312–314.

[9] S. Grillner, J.T. Buchanan, P. Wallén, L. Brodin, Neural control of locomotion in lower vertebrates, in: A.H. Cohen, S. Rossignol, S. Grillner (Eds.), Neural Control of Rhythmic Movements in Vertebrates, John Wiley & Sons, New York, 1988, pp. 1–40.

[10] S. Grillner, P. Wallén, L. Brodin, Neuronal network generating locomotor behavior in lamprey: circuitry, transmitters, mem-

brane properties, and simulation, Ann. Rev. Neurosci. 14 (1991) 169–199.

[11] S. Grillner, T. Degliana, Ö. Ekeberg, A. El Marina, A. Lansner, G. Orlovsky, P. Wallén, Neural networks that co-ordinate locomotion and body orientation in lamprey, Trends Neurosci. 18 (6) (1995) 270–279.

[12] K.L. Paap, M. Dehlwisch, B. Klaassen, GMD-Snake: a semi-autonomous snake-like robot, Distributed Autonomous Robotic Systems 2, Springer-Verlag, 1996.

[13] B. Klaassen, K.L. Paap, GMD-SNAKE2: a snake-like robot driven by wheels and a method for motion control, in: ICRA 1999: Proceedings of 1999 IEEE International Conference on Robotics and Automation, IEEE, 1999, pp. 3014–3019.

[14] H.R. Choi, S.M. Ryew, Robotic system with active steering capability for internal inspection of urban gas pipelines, Mechatronics 12 (2002) 713–736.

[15] Y. Umetani, S. Hirose, Biomechanical study of active cord mechanism with tactile sensors, Proceedings of the 6th International Symposium on Industrial Robots, Nottingham, 1976 c1-1–c1-10.

[16] S. Hirose, Biologically Inspired Robots (Snake-like Locomotor and Manipulator), Oxford University Press, 1993.

[17] G.S. Chirikjian, J.W. Burdick, Design, implementation, and experiments with a thirty-degree-of-freedom 'hyper-redundant' robot, in: ISRAM 1992.

[18] T. Lee, T. Ohm, S. Hayati, A highly redundant robot system for inspection, in: Proceedings of the Conference on Intellingent Robotics in the Field, Factory, Service, and Space (CIRFFSS'94), Houston, TX, 1994, pp. 142–149.

[19] G.S.P. Miller, J. Ayers, J.L. Davis, A. Rudolph, Neurotechnology for Biomimetic Robots, Bradford/MIT Press, Cambridge London, 2002 (Ch. Snake robots for search and rescue).

[20] M. Saito, M. Fukaya, T. Iwasaki, Serpentine locomotion with robotic snakes, IEEE Control Syst. Magazine 22 (2002) 64–81.

[21] J. Conradt, P. Varshavskaya, Distributed central pattern generator control for a serpentine robot, in: ICANN, 2003.

[22] R. Worst, Robotic Snakes, Third German Workshop on Artifical Life Verlag Harri Deutsch 1998, pp. 113–126.

[23] K.A. McIsaac, J.P. Ostrowski, A geometric approach to anguilliform locomotion: simulation and experiments with an underwater eel-robot, in: ICRA 1999: Proceedings of 1999 IEEE International Conference on Robotics and Automation, IEEE, 1999, pp. 2843–2848.

[24] C. Wilbur, W. Vorus, Y. Cao, S.N. Currie, J. Ayers, J.L. Davis, A. Rudolph, Neurotechnology for Biomimetic Robots, Bradford/MIT Press, Cambridge London, 2002 (Ch. A Lamprey-Based Undulatory Vehicle).

[25] T. Takayama, S. Hirose, Development of HELIX: a hermetic 3D active cord with novel spiral swimming motion, in: Proceedings

of TITech COE/Super Mechano-Systems Symposium, 2001, pp. D-3.

[26] S. Hirose, E.F. Fukushima, Snakes and Strings: New Robotic Components for Rescue Operations, in: B. Siciliano, D. Paolo (Eds.), Experimental Robotics VIII: Proceedings of the 8th International Symposium ISER02, Springer-Verlag, 2002, pp. 48–63.

[27] T. Takayama, S. Hirose, Amphibious 3D active cord mechanism "HE-LIX" with Helical swimming motion,in: Proceedings of the 2002 IEEE/RSJ International Conference on Inteligent Robots and Systems, IEEE, 2002, pp. 775–780.

[28] S. Ma, H. Araya, L. Li, Development of a creeping snake-robot, in: Proceedings of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation. IEEE, 2001, pp. 77–82.

[29] J. Ostrowski, J. Burdick, Gait kinematics for a serpentine robot, Proceedings of the 1996 IEEE International Conference on Robotics and Automation, IEEE, 1996, pp. 1294–1299.

[30] F. Matsuno, K. Suenaga, Control of redundant 3D snake robot based on kinematic model, Proceedings of the 2003 IEEE International Conference on Robotics & Automation, IEEE, 2003, pp. 2061–2066.

[31] P. Prautsch, T. Mita, Control and analysis of the gait of snake robots, Proceedings of the 1999 IEEE International Conference on Control Applications, IEEE, 1999, pp. 502–507.

[32] H. Date, Y. Hoshi, M. Sampei, N. Shigeki, Locomotion control of a snake robot with constraint force attenuation, Proceedings of the American Control Conference, AACC, 2001, pp. 113–118.

[33] J. Ute, K. Ono, Fast and efficient locomotion of a snake robot based on self-excitation principle, Proceedings of the 7th International Workshop on Advanced Motion Control, IEEE, 2002, pp. 532–539.

[34] A.J. Ijspeert, J. Hallam, D. Willshaw, Evolving swimming controllers for a simulated lamprey with inspiration from Neurobiology, Adaptive Behav. 7 (2) (1999) 151–172.

[35] A.J. Ijspeert, A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander, Biol. Cybernet. 85 (5) (2001) 331–348.

[36] A.J. Ijspeert, J.-M. Cableguen, Gait transition from swimming to walking: investigation of salamander locomotion control using nonlinear oscillators, in: Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines, 2003.

[37] O. Michel, Webots: Professional mobile robot simulation, J. Adv. Robot. Syst. 1 (1) (2004) 39–42.

[38] J. Braure, O. Michel, A. Crespi, A.J. Ijspeert, Exploration of the morphological configuration and the locomotion controller for a simulated amphibious salamander robot, 2004, in preparation.