An amphibious robot capable of snake and lamprey-like locomotion

Alessandro Crespi, André Badertscher, André Guignard, Auke Jan Ijspeert
School of computer and Communication Sciences
EPFL, Swiss Federal Institute of Technology, Lausanne
CH-1015 Lausanne, Switzerland
Contacts: alessandro.crespi@epfl.ch, auke.ijspeert@epfl.ch

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 the lamprey in water and serpentine 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. Preliminary results in simulation and with the first elements that compose the real robot are presented.

Introduction

This project aims at constructing a biologically inspired amphibious snake-like robot. 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 so-called *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 next section), the main features of our robot are (1) to be amphibious and capable of both swimming and serpentine 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 related work. We will then describe the design considerations underlying the project, followed by a detailed description of the hardware and software of the robot. Preliminary results in simulation and with the real robot are then described. We finish the article with a description of future work and a short conclusion.

Related snake-like robots

Most of the snake robots that exist actually have been developed for the following two main purposes:

- The inspection of pipes (particularly sewage pipes) that are not easily accessible by humans. These robots are generally quite big and slow; their motion is generally artificial as it is often based on powered wheels. They clearly need to be waterproof.
- The implementation of some types of snake locomotion. These robots are generally smaller and faster than those designed for the inspection of pipes, and implement real snake gaits.

One of the first snake robot concept is the *Active Cord Mechanism* of Shigeo Hirose, cited with some others snake robots he built in (Hirose, 1993). Not many robots are actually capable of having realistic snake-like motion. Two of the most realistic ones are probably *S5* (Miller, 2002) and *Snake2* (Klaassen & Paap, 1999). Both robots are not waterproof.

The swimming snake-robots are often designed to imitate the anguilliform swimming gaits of the eel (or the very similar ones of the lamprey). Many 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* (McIsaac & Ostrowski, 1999) and the lamprey robot built at Northeastern University (Ayers, Wilbur, & Olcott, 2000).

To the best of our knowledge there is currently no snake-like robot that can both swim in water and crawl with serpentine locomotion on ground.

Design considerations

The robot 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 inactive. 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 serpentine locomotion (lower friction coefficient in the longitudinal axis compared to the perpendicular axis).

• to be controlled by a CPG composed of coupled nonlinear 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.

Hardware

The robot is modular and constructed out of several identical segments, named *elements* (Figure 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 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.

Figure 1. Two connected elements

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 some 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 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 are 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 two hours (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 one hour.

The motor controller is built with a PIC microcontroller (PIC 16F876) and some external components. The motor has a magnetic encoder that generates 16 impulsions for every complete rotation of the axis. This encoder is connected to a LS 7084 quadrature detector that decodes the signals of the magnetic coder, 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 analogical 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 coder 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 the speed of the motor to be changed.

Between the H-bridge and the motor, a 1 Ω 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 analogical 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 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 0.5 Hz can be obtained if the full amplitude (60°) is used.

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 I^2C bus and the external power source all along the robot. The fifth wire is currently unused and is reserved for future applications.

The tail of the robot is empty: being the last element, it doesn't need a motor, nor any controller or power supply. The first element (head) 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 (power supply, PC interface).

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. Central pattern generators (CPGs) are networks of neurons that can produce coordinated oscillatory signals without oscillatory inputs (Delcomyn, 1980). 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 (Buchanan & Grillner, 1987; Grillner, Buchanan, Wallén, & Brodin, 1988; Grillner, Wallén, & Brodin, 1991; Grillner et al., 1995); 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.

In previous work, we have modelled CPGs for swimming and walking using neural network simulations (Ijspeert, Hallam, & Willshaw, 1999; Ijspeert, 2001). Here we will instead use nonlinear oscillators as building blocks for constructing CPGs. 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.

We use the following nonlinear oscillator:

$$\left\{ \begin{array}{rcl} \tau\dot{v} & = & -\alpha\dfrac{x^2+v^2-E}{E}v - x \\ \tau\dot{x} & = & v \end{array} \right. \tag{1}$$

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 amplitude $\sqrt{E}$ 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. 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.

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:

$$\left\{ \begin{array}{rcl} \tau\dot{v}_i & = & -\alpha\dfrac{x_i^2+v_i^2-E_i}{E_i}v_i - x_i \\ & & +\sum_j \left(a_{ij}x_j + b_{ij}v_j\right) \\ \tau\dot{x}_i & = & v_i \end{array} \right. \tag{2}$$

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

The CPG used in this project is composed of a chain of oscillators (Figure 2, left). For simplicity, we assume that only nearest neighbor connections exist between oscillators, and that each oscillator has the same coupling weights to its rostral and caudal neighbors. By exploring the parameter space of different possible coupling weights between oscillators, it is easy to find couplings that produce travelling waves from head to tail necessary for swimming and anguilliform swimming (see next section).

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.

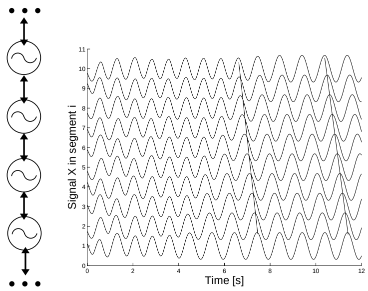

Figure 2. Left: Configuration of the body CPG. Right: Oscillations in a 10-oscillator chain. The oblique lines show that a travelling wave with a wavelength of approximately the length of the chain is obtained.

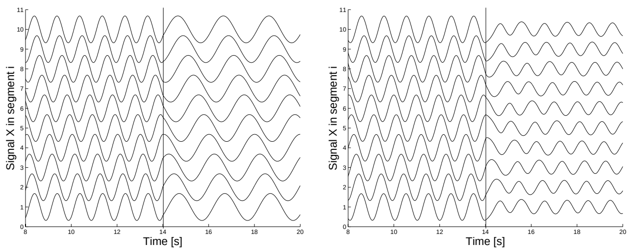

Figure 3. Left: Modulation of the period by doubling the parameters τ_i at time t=14s. Right: Modulation of the amplitude by dividing the parameters E_i by a factor 4 at time t=14s.

Results

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. Figure 2 (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 $(x_i, v_i) = (0,0)$ for all i, which is an unstable fixed point).

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. Figure 3 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 adjust the speed of locomotion depending on the conditions.

One of the main motivation for using nonlinear oscillators, is their ability to cope with transient perturbations. When correctly coupled a chain of oscillators produces a

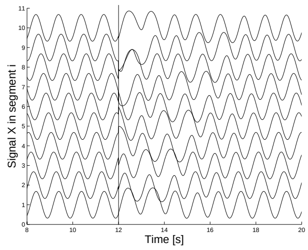

Figure 4. Random perturbation of the state variables x_i at time t=12s.

stable limit cycle behavior to which the system will evolve from any initial conditions and after any type of transient perturbation (except from the unstable fixed point mentioned above). Figure 4 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 returns to the original travelling wave.

The locomotion controller is currently being tested both in a dynamic simulation, and with the first elements for the real robot. Using Webots Dynamics, 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 (Figure 5). We are currently exploring which types of travelling waves (in terms of wavelength and amplitude of oscillation) produce the fastest locomotion gaits (both serpentine and anguilliform), for a given frequency of oscillation. We are also investigating in simulation how the number of elements in the robot affects its speed of locomotion. Preliminary results show that good swimming and serpentine gaits can be obtained. Differences between optimal waves in water and on ground are currently being investigated.

The real robot is currently made of four elements, which is not yet sufficient for either swimming or serpentine locomotion. A robot made of up to 15 elements will be ready in a very near future. We extensively tested the frequencies and amplitudes of oscillations that the elements can deliver (with and without load), and it appears the robot will be able to swim and crawl with the maximum amplitude of oscillations (60 degrees) at 0.5 Hz.

Future work

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

- 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 achieve a real autonomy.

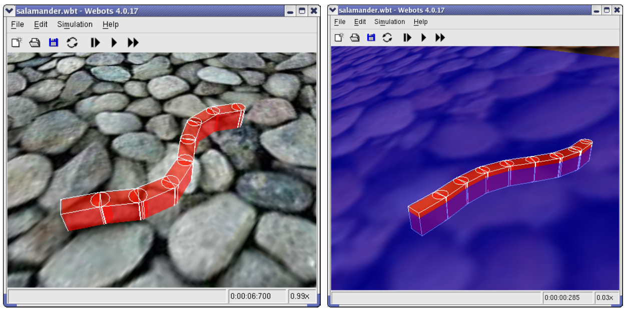

Figure 5. Two screenshots of the simulation. Left: Simulated snake robot on ground. Right: Simulated snake robot in water.

• 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.

• 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 serpentine locomotion on the ground are still to be analyzed in detail. As asymmetric friction is required for this type of locomotion, the ways to obtain it 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 concertina 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 (Ijspeert, 2001). We are currently developing special elements with simple limbs to add walking as an additional available gait.

Conclusions

This article presented the first developments in a project that aims at developing an amphibious snake-like robot capable of swimming and serpentine 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.

Acknowledgements We gratefully acknowledge the support of the following people:

• Francesco Mondada and the Autonomous Systems Laboratory (ASL) at the EPFL, for their PD motor controller;

• Daniel Burnier, for providing the RS-232 – I^2C interface;

• Simon Capern, Anurag Tripathi, and Jérôme Braure for helping with the development of the simulation of the robot in Webots;

• Olivier Michel for adjusting Webots to our needs.

References

Ayers, J., Wilbur, C., & Olcott, C. (2000). Lamprey robots. In *Proceedings of the international symposium on aqua biomechanisms.* Tokai University.

Buchanan, J., & Grillner, S. (1987). Newly identified 'glutamate interneurons' and their role in locomotion in the lamprey spinal cord. *Science*, *236*, 312-314.

Delcomyn, F. (1980). Neural basis for rhythmic behaviour in animals. *Science*, *210*, 492-498.

Grillner, S., Buchanan, J., Wallén, P., & Brodin, L. (1988). Neural control of locomotion in lower vertebrates. In A. H. Cohen, S. Rossignol, & S. Grillner (Eds.), *Neural control of rhythmic movements in vertebrates* (p. 1-40). Jon Wiley & Sons.

Grillner, S., Degliana, T., Ekeberg, Ö., El Marina, A., Lansner, A., Orlovsky, G., & Wallén, P. (1995). Neural networks that coordinate locomotion and body orientation in lamprey. *Trends in Neuroscience*, *18*(6), 270-279.

Grillner, S., Wallén, P., & Brodin, L. (1991). Neuronal network generating locomotor behavior in lamprey: Circuitry, transmitters, membrane properties, and simulation. *Annu. Rev. Neurosci.*, *14*, 169-199.

Hirose, S. (1993). *Biologically inspired robots (snake-like locomotor and manipulator).* Oxford University Press.

Ijspeert, A. (2001). A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, *85*(5), 331-348.

Ijspeert, A., Hallam, J., & Willshaw, D. (1999). Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology. *Adaptive Behavior*, *7*(2), 151-172.

Klaassen, B., & Paap, K. (1999). 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* (p. 3014-3019). IEEE.

McIsaac, K., & Ostrowski, J. (1999). 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* (pp. 2843–2848). IEEE.

Miller, G. (2002). In J. Ayers, J. Davis, & A. Rudolph (Eds.), *Neurotechnology for biomimetic robots.* Bradford/MIT Press, Cambridge London.