Salamandra robotica: a biologically inspired amphibious robot that swims and walks

Alessandro Crespi, Auke Jan Ijspeert

1 Introduction

One of the key characteristics of animals, perhaps the most impressive, is their ability to move. It is the result of millions of years of evolution, and its complexity, flexibility and energy efficiency are yet to be approached by robots. The control and coordination of many degrees of freedom in a robot is complex, and there is not well established technique to deal with this: on their side, animals often have hundreds degrees of freedom and use them with a surprising ability.

The purpose of this project is to develop an amphibious robot whose structure and control are inspired by the salamander in two aspects: the biomechanical structure and the locomotion control. The first purpose of the project is to explore and develop new technologies inspired by the salamander. In particular, we aim at developing a robot that can robustly swim, crawl, and walk. The second purpose is to use the developed robot as a test-bed for neurobiological models in a real (as opposed to simulated) embodiment. Finally, with its multiple gaits, such a robot would be useful for inspection or exploration purposes in difficult environments (e.g., flooded zones, under collapsed buildings, etc.).

The salamander, a tetrapod capable of both swimming and walking, offers a remarkable opportunity to investigate vertebrate locomotion. It is considered as closely resembling the first terrestrial vertebrates and represents, among vertebrates, a key animal in the evolution from the aquatic to the terrestrial environment [6, 18]. It also has orders of magnitudes fewer neurons than mammals [29, 30] and is therefore at a level of complexity that is more tractable from the understanding and modeling point of views. Finally, the central nervous system of the salamander shares many similarities with that

Biologically Inspired Robotics Group, Ecole Polytechnique Fédérale de Lausanne, Station 14, CH-1015 Lausanne

of the lamprey, an extensively studied primitive fish, and many data and models of the lamprey's swimming circuitry are therefore available to guide the understanding of the salamander's locomotor circuitry. From a robotics point of view, it is very attractive to develop an amphibious robot capable of swimming, crawling, and walking. To the best of our knowledge, such a robot has never been developed before.

1.1 Robots as tools for biology

Robots are increasingly being used as tools to verify biological hypotheses or as models of biological sensorimotor systems [37]. Examples include lamprey locomotion [41, 33], lobster locomotion [1], cricket phonotaxis [40] and cat locomotion [17]. For a more detailed review, see [38, 39].

Compared to computer simulations, the use of real robots is interesting as it provides several advantages:

- The model is completely interacting with a real environment, using real sensors and real actuators. This therefore eliminates the need to simulate the sensors and the actuators (which can be generally simulated only with approximate models). The absence of simplified models or biased results owing to the simulation is a great advantage, as some aspects could strongly depend on the interaction with the environment.
- There is no need to simulate complex environments or complicated force models. A correct simulation of some phenomena (for example friction forces, hydrodynamical forces, etc.) is extremely difficult (especially if associated with articulated moving bodies, whose shape is not constant); simulations are therefore generally limited to a simplified model which could introduce artifacts that cause the model to behave differently than in the real world.

However, it is also important to notice that the use of real robots has some drawbacks compared to simulations:

- Reproducing the mechanical properties of real animal bodies on a robot is very difficult. A robot will only be an approximation of the real animal, as generally it is technically not feasible to build a robot having the same properties (especially for the number of degrees of freedom: a robot with hundreds of degrees of freedom like a real snake would be much larger than the animal with the current technology). The visco-elastic properties of animal muscles are also difficult to implement in robots.
- Almost everything can be designed in a simulation, including systems using components (e.g., sensors or actuators) that are expensive, hard to use, or even not existing with the current technology. For example, some animal sensor systems (like touch) are difficult to replicate with currently existing sensing devices.

- Building a robot generally requires much more work than implementing a simulation, and robots sometimes have to be repaired or maintained. Moreover, robots only run in real time (simulations can be faster).

1.2 Related work

This section briefly presents existing snake and salamander robots.

1.2.1 Snake-like robots

Snake-like robots can be classified into two main groups:

- Robots that move using powered wheels or caterpillars (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 [27, 23] and are sometimes used, for example, for the inspection of pipes [5]. 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.

One of the first known snake robots was built by Hirose and colleagues at the end of 1972 [36]. He generically named this kind of robot an *active cord mechanism* (ACM). After this first prototype he built some other snake robots [20]. A huge snake robot has been developed in 1992 at Caltech [4]. The Jet Propulsion Laboratory of the NASA presented in 1994 a serpentine robot [24]. Miller developed several prototypes of snake robots; among them the last one, S5 [26], 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 colleagues presented in 2002 a simple snake robot used to validate some theoretical results [31]. Conradt developed WormBot [7], a snake-like robot controlled by local central pattern generators (CPGs). For a more detailed review of snake-like robots, see [14, 42].

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 [25], the lamprey robot built at Northeastern University [41], and the lamprey-like robot built at SSSA [33]. 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 are currently only a few amphibious snake-like robots, the HELIX-I ([34] as cited in [21]; [35]) and its successor ACM-R5 [43], that can both swim in water and crawl on the ground (although ground locomotion is not described in the papers).

1.2.2 Salamander-like robots

Currently only a few prototypes of salamander-like robot (i.e., quadruped robots with several degrees of freedom in the spine) have been object of scientific publications:

- A salamander robot with 6 segments and an on-board FPGA-based control system has been presented in [19]. It is not amphibious and can only walk.
- Robo-Salamander, a salamander robot with two degrees of freedom for the spine, and two for each leg, has been presented in [2]; no experiments seem to have been done with it, and no other publications followed. This robot was not autonomous and was powered and controlled using a cable. It is only capable of walking.

There are also some legged robots with flexible spine built by hobbyists, whose descriptions can be found on Internet, but none of them has been designed or used for scientific experiments.

None of the robots listed here is capable of swimming, and none is fully autonomous or amphibious.

1.3 Central pattern generator model

The swimming motion of salamanders is similar to the one of lampreys, using axial undulations which propagate as travelling waves from head to tail. The walking motion has a different pattern: the salamander moves the diagonally opposed limbs together, generating at the same time an S-shaped standing wave (which has nodes at the girdles) with the body.

Using the salamander as model, we address three fundamental issues related to vertebrate locomotion: (i) the modifications of the spinal locomotor circuits during the evolutionary transition from aquatic to terrestrial locomotion, (ii) the mechanisms needed for the coordination of limb and body (i.e., axial) movements, and (iii) the mechanisms that underlie gait transitions induced by simple electrical stimulation of the brainstem.

Our model is based on the following hypotheses:

1. The body CPG is like that of the lamprey and spontaneously produces travelling waves when activated with a tonic drive. The limb CPG, when activated, forces the whole CPG into the walking mode, as previously proposed in [6].
2. The strengths of the couplings from limb to body oscillators are stronger than those from body to body oscillators and from body to limb oscillators. This allows the limb CPG to "override" the natural tendency of the body CPG to produce travelling waves and force it to produce standing waves.
3. Limb oscillators can not oscillate at high frequencies, that is, they saturate and stop oscillating at high levels of drive. This provides a mechanism for automatically switching between walking and swimming when the drive is varied [3], and explains why swimming frequencies are systematically higher than walking frequencies [16, 13].
4. For the same drive level, the intrinsic frequencies of the limb oscillators are lower than those of the body oscillators. This explains the rapid increase of frequency during the switch from walking to swimming and the gap between the walking and swimming frequency ranges [16, 13].

More details about the underlying biological hypotheses can be found in [22].

The body CPG model is a double chain of oscillators with nearest neighbor coupling (Fig. 1). An oscillator models the activity of an oscillatory center in the spinal cord (a group of about 50000 neurons that produce rhythmic activity). The chain is designed to generate a travelling wave, from the head to the tail of the robot. This wave is used to achieve anguilliform swimming in water. In addition to this body CPG, limb oscillators have been added to the model (one per limb); they are bidirectionally coupled together and unidirectionally coupled to all body oscillators (see Fig. 1). During swimming, these oscillators are stopped (they do not oscillate), and thus do not influence the behaviour of the body CPG, which continues to produce a travelling wave. During walking, the oscillators are enabled and influence the body oscillators, which begin to produce an S-shaped standing wave that can be used for walking.

The total number of oscillators is $N = 20$: $N_B = 16$ oscillators (i.e., 8 pairs) for the body CPG (which controls 6 real elements and 2 fictive joints placed in the limb elements), and $N_L = 4$ oscillators for the limbs. Body joints (both real and fictive) are numbered 1 to 8 from head to tail. Oscillators in the left chain of the CPG are numbered 1 to 8 and those on the right side are numbered 9 to 16 from head to tail. Limb oscillators are numbered 17 to 20.

The oscillators that compose the CPG are implemented as amplitude-controlled phase oscillators:

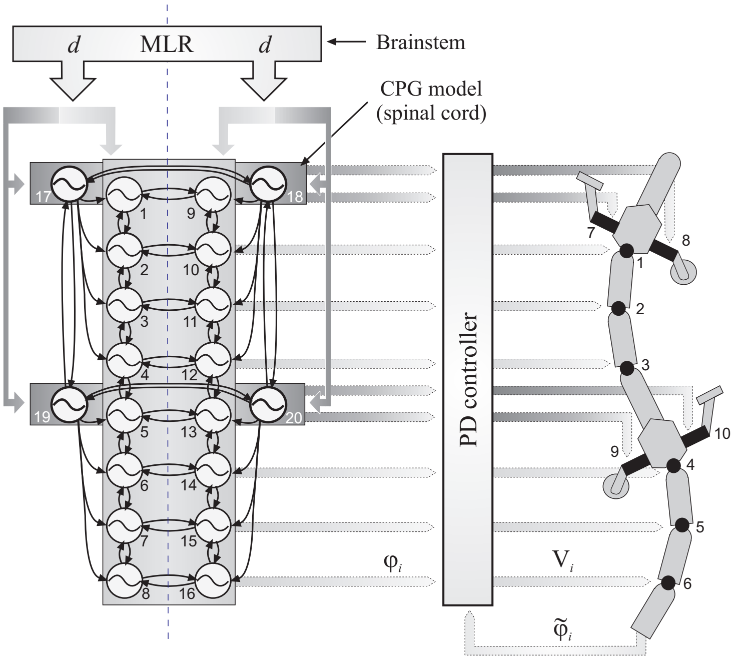


Fig. 1: Structure of the CPG used in the robot

$$\begin{cases} \dot{\theta}_i &= 2\pi\nu_i + \sum_j w_{ij} \sin\Big(\theta_j - \theta_i - \phi_{ij}\Big) \\ \ddot{r}_i &= a_i\Big(\frac{a_i}{4}(R_i - r_i) - \dot{r}_i\Big) \\ x_i &= r_i\Big(1 + \cos(\theta_i)\Big) \end{cases} \tag{1}$$

where the state variables θ_i and r_i represent, respectively, the phase and the amplitude of the i^{th} oscillator, the parameters ν_i and R_i determine the intrinsic frequency and amplitude, and a_i is a positive constant. The coupling between the oscillators is defined by the weights w_{ij} and the phase biases ϕ_{ij}. The variable x_i is the rhythmic and positive output signal extracted out of oscillator i.

1.3.1 Influence of drive: saturation function

To allow the control of the whole CPG with a single parameter, reproducing the output of the mesencephalic locomotor region (MLR) in the animal (i.e., the region of the brainstem that outputs the descending signals controlling the locomotion), a *saturation function* has been introduced:

$$\begin{pmatrix} \nu_i \\ R_i \end{pmatrix} = g(d) = \begin{pmatrix} g_\nu(d) \\ g_R(d) \end{pmatrix} \tag{2}$$

This function is a stepwise linear function, defined by the following equations:

$$g_\nu(d) = \begin{cases} c_{\nu,1}d + c_{\nu,0} & \text{if } d_{\text{low}} \le d \le d_{\text{high}} \\ \nu_{\text{sat}} & \text{otherwise} \end{cases} \tag{3}$$

$$g_R(d) = \begin{cases} c_{R,1}d + c_{R,0} & \text{if } d_{\text{low}} \le d \le d_{\text{high}} \\ R_{\text{sat}} & \text{otherwise} \end{cases} \tag{4}$$

Body and limb oscillators use different saturations functions, as they have to saturate at different levels of drive (see Fig. 3).

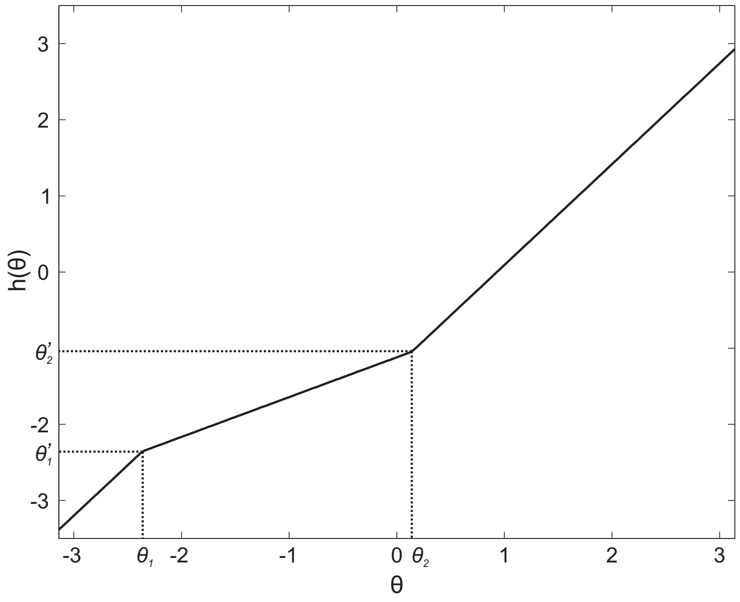


Fig. 2: Leg rotation function used on the robot. θ_1 and θ_2 correspond, respectively, to the begin and end of the stance phase of the leg

The frequency and amplitude parameters of all oscillators are determined, on the base of the input drive d, by the saturation function:

$$\begin{aligned} \begin{pmatrix} \nu_{\text{body}} \\ R_{\text{body}} \end{pmatrix} &= g_{\text{body}}(d) \\ \begin{pmatrix} \nu_{\text{limb}} \\ R_{\text{limb}} \end{pmatrix} &= g_{\text{limb}}(d) \end{aligned} \tag{5}$$

The actual saturation function is plotted in Fig. 3.

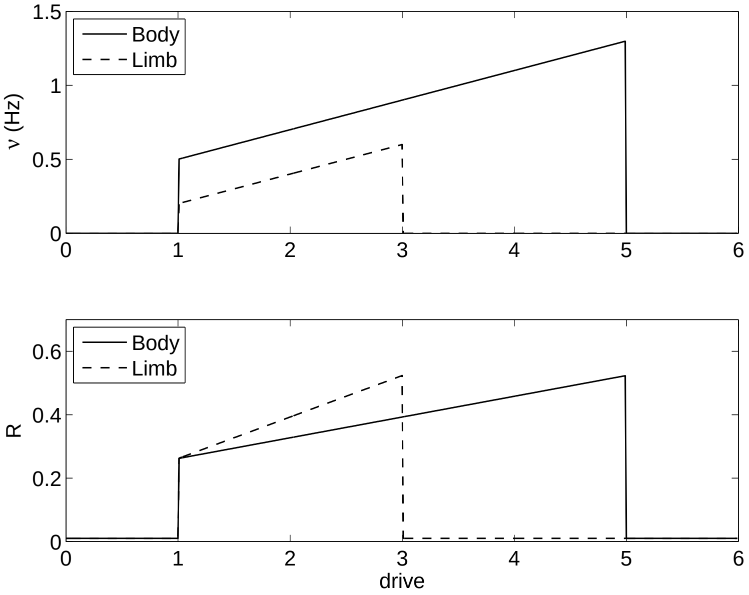


Fig. 3: Saturation function used on the robot

To achieve turning, different drives d_L and d_R can be applied to the left and right sides of the body oscillator.

1.3.2 Setpoints

For the body, the setpoints φ_i, i.e., the desired angles for the 6 actuated joints, are obtained by taking the difference between the x_i signals from the left and right oscillators. A standard PD motor controller is then used to compute the voltage τ_i (i.e., torque) applied to the motor (using a PWM signal):

$$\begin{aligned} \varphi_i &= x_i - x_{N+i} \\ \tau_i &= K_p e_i + K_d \dot{e}_i \end{aligned} \tag{6}$$

where $e_i = \varphi_i - \tilde{\varphi}_i$ is the tracking error between the desired angles φ_i and the actual angles $\tilde{\varphi}_i$ measured by the motor incremental encoders, and K_p and K_d are the proportional and derivative gains.

Because the limbs need to make complete rotations, their setpoints should monotonically increase, instead of having rhythmic movements like the body joints. The setpoints φ_i are therefore directly calculated from the phases θ_i of the limb oscillators using a non-uniform rotation function (which accelerates the movement of the leg when it is not in contact with the ground):

$$\begin{aligned}\varphi_i &= h(\theta_i)\\ \tau_i &= K_p e_i + K_d \dot{e}_i\end{aligned} \tag{7}$$

The $h(\theta)$ function used on the robot is plotted in Fig. 2. With this function, the stance duration is approximately 40% of a whole cycle.

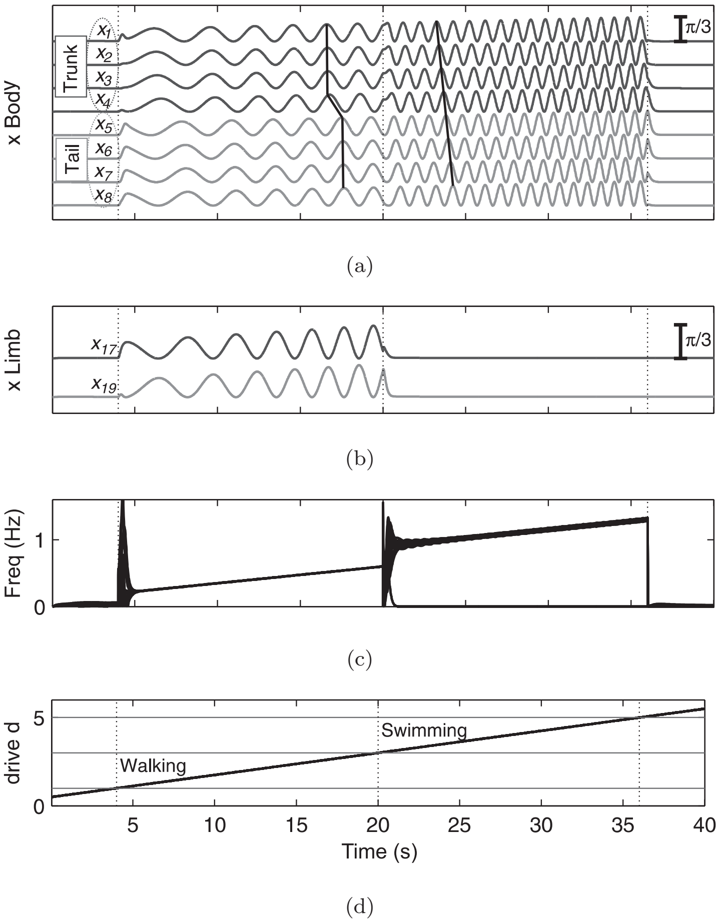
1.3.3 Interoscillator couplings

In the body oscillator, the phase biases ϕ_{ij} are chosen to be equal to π between left and right oscillators (i.e., these will oscillate in anti-phase). The phase biases between neighbor oscillators are set to $\frac{2\pi}{N_B/2} = \frac{2\pi}{8}$ for the descending connections and to $-\frac{2\pi}{N_B/2} = -\frac{2\pi}{8}$ for the ascending connections; this produces a complete wave of the body. In the limb oscillator and for the couplings between body and limb oscillators, $\phi_{ij} = \pi$ is used for all connections.

We use $w_{ij} = 10$ for all connections, with the exception of the couplings from limb to body oscillators, which need to be stronger, for which a value of $w_{ij} = 30$ has been used. For all oscillators, $a_i = 20$. The PD coefficients K_p and K_d are tuned manually for each element (e.g., elements in middle of the chain require larger gains than those at the extremities for good trajectory tracking).

The d parameter can be modified online by a human operator from a control PC using the wireless connection. The CPG will rapidly adapt to any parameter change and converge to the modified travelling or standing wave after a short transient period. An example of how the CPG reacts to parameter changes can be observed in Fig. 4: even with a continuously changing input drive, the oscillator generates smooth trajectories without any discontinuities in the outputs.

The differential equations are integrated by the microcontroller of the head (see Sect. 3.3) using the Euler method, with a time step of 10 ms and using fixed point arithmetics. As the current trajectory generator has a limited computing power (10 MIPS with one 8-bit register), the code heavily uses lookup tables for calculating functions.



(a)

(b)

(c)

(d)

Fig. 4: Switching from walking to swimming; activity of the CPG model when the drive signal is progressively increased. **a** x_i signals from the left body CPG oscillators (oscillators on the right side are exactly in antiphase). Units are in radians (scale bar on the top right). Note the transition from standing waves (with synchrony in the trunk, synchrony in the tail, and an antiphase relation between the two, $4\,\mathrm{s} < t < 20\,\mathrm{s}$) to travelling waves ($20\,\mathrm{s} < t < 36\,\mathrm{s}$). **b** x_i signals from the left-limb CPG oscillators. Ipsilateral fore- and hindlimbs are in antiphase. **c** Instantaneous frequencies measured as $\frac{\dot{\theta}_i}{2\pi}$ in cycles/s. The variations in the instantaneous frequencies among individual oscillators at times $t = 4\,\mathrm{s}$ and $t = 20\,\mathrm{s}$ correspond to brief accelerations and decelerations before resynchronization. **d** Linear increase of the drive d applied to all oscillators. The horizontal lines correspond to the lower ($d_{\mathrm{low}}^{\mathrm{limb}} = d_{\mathrm{low}}^{\mathrm{body}} = 1$) and upper ($d_{\mathrm{high}}^{\mathrm{limb}} = d_{\mathrm{high}}^{\mathrm{body}} = 5$) oscillation thresholds for limb and body oscillators in arbitrary drive units

2 Robot's design

2.1 First prototype

The first prototype of salamander robot (Fig. 5) was built using three body elements and two limb elements. A head element was also present, but was internally empty, the trajectory generator being an offboard computer connected to the robot with a shielded cable. The body elements were those of the AmphiBot I snake robot [8, 9, 10].

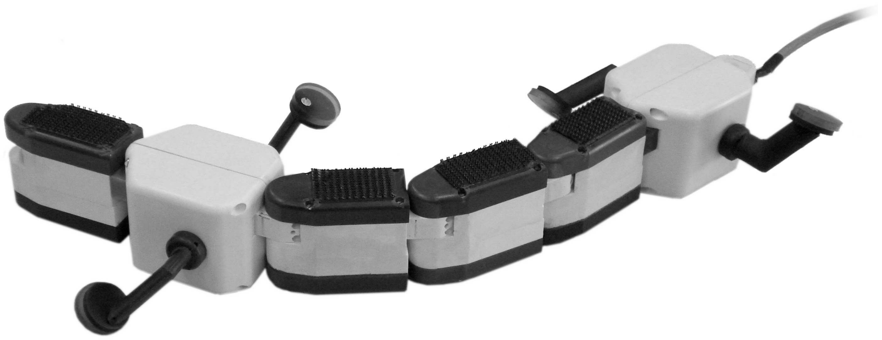
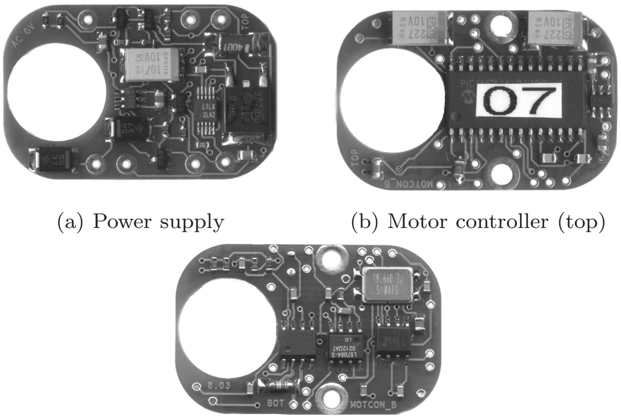


Fig. 5: The prototype of salamander robot built with the first generation elements

2.2 Body elements

Each element has a single degree of freedom, and elements are fixed such that all axes of rotation are aligned. They consist of four structural parts: a body, two covers and a connection piece (a drawing of two connected elements is visible in figure 7). All parts are molded using polyurethane, using molds created from positive parts in aluminium shaped with a CNC milling machine. 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 (the bottom O-ring has been subsequently replaced by a silicone sealant, because the complete closing of the bottom cover was generating mechanical problems to the gearbox). An element has a length of 7 cm and a section of 5.5 by 3.3 cm.

Each element contains two printed circuits (one for the power supply/battery charger and one for the motor controller, see figure 6), 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 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 (LTC3200). 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 (LTC1733) is part of the power supply circuit. The charger can be enabled or disabled by the user over the I^{2}C bus. The battery has a capacity of 600 mAh, which is enough to power the element for an average time of approximately two 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 one hour.



(a) Power supply

(b) Motor controller (top)

(c) Motor controller (bottom)

Fig. 6: Pictures of the printed circuit boards of the first prototype (real size)

The motor controller is built with a PIC microcontroller (PIC16F876) 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 LS7084 quadrature detector that filters and decodes the

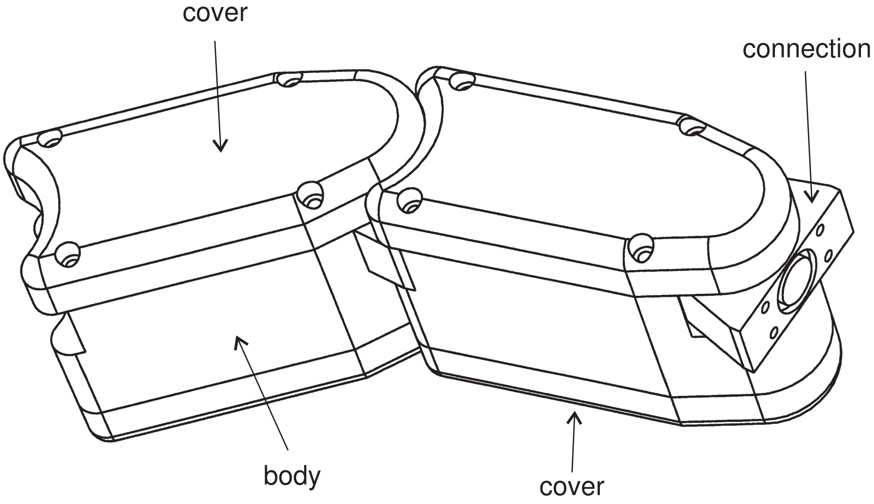


Fig. 7: Drawing of two first generation elements connected together

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 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 coder and the real one).

The motor coil is powered through a SI9986 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 INA146 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 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 0.3 Hz can be obtained if the full amplitude (±45°) 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.

2.3 Limb elements

The first limb elements, very similar in structure to the current ones (see section 3.2) but with the same electronics as the old body elements, were also tested as a first prototype of a salamander robot (without its tail; see figure 5).

Most of the body elements were damaged by water leakages (which were not immediately detected) after the tests published in [10]; only three of them were still working and have been used for the salamander prototype. This was therefore only a preliminary design for testing the conception of the limb elements (no experiments have been done), and no detailed description of them will be given.

2.4 Design problems

This first prototype suffered of several design problems, which have been mostly corrected in the current version of the elements (see next sections):

- The direct use of 5 V for the external power supply (mostly due to the lack of internal space for step-down converters, which require big coils) rendered the usage of the robot with external power (and the battery charging) very problematic, as only a limited amount of current can pass through the internal wires (having a section of $0.127\,\mathrm{mm}^2$). For instance, a current of approx. 2 A on the wires caused a voltage drop along the robot around 2.5 V, causing part of the elements to reset (disabling battery charging).
- The torque generated by the elements was insufficient to achieve full oscillations at frequencies greater than 0.3 Hz, resulting in very slow locomotion.
- The waterproofing of the elements was very problematic and required sealing them with silicone.
- There was no possibility to detect the presence of water inside elements. Any malfunctioning supposedly owing to water leakage required the robot to be completely unmounted.
- The rigid connection between the elements combined with the small differences in the pieces caused the mounted robot to have bad contacts with the ground (i.e., it was not perfectly flat).
- No battery protection mechanisms were implemented, and there was no possibility of turning off the robot, therefore it had to remain connected

to the external power all the time to preserve the batteries from being completely discharged (and thus rendered unusable).

- No connectors were on the circuits, and all the connections (including those to the motor) were realized by directly soldering the wires to the PCB. This operation was difficult (hence giving high mounting times for each element) and rather unreliable.
- The absence of any onboard trajectory generation capabilities and of radio communication required the direct control of the robot through a long shielded cable connected to a PC using a RS-232–I^{2}C converter.

3 Hardware

This new prototype (Fig. 8) address most of the problems found with the previous one, particularly in terms of mounting simplicity, electronic reliability, and waterproofing.

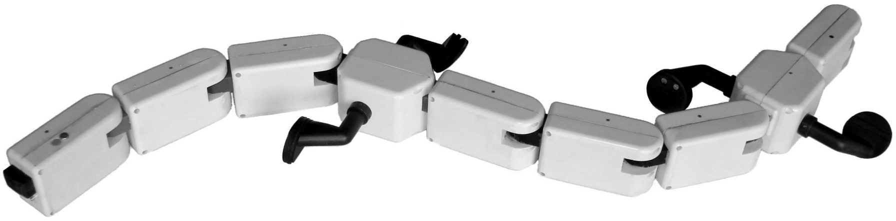


Fig. 8: Salamandra robotica

3.1 Body elements

The same material of the first generation elements (polyurethane resin lighted with glass microballs) has been chosen for the external casing of the elements. They consist of two vertical symmetrical parts that are fixed together with screws. This is different from the first prototype which was having a body closed with two covers (top and bottom). The elements are connected (both mechanically and electrically) using a compliant connection piece (molded with polyurethane rubber) fixed to the output axis, which contains 6 wires. The use of compliant connection pieces corrects the bad contact with the ground that was a serious problem of the previous generation elements, and allows the robot to better deal with irregularities of the ground. All the output axes of the elements are aligned, therefore producing planar locomotion. To ensure the waterproofing of each element, custom O-rings (placed between the two parts composing the body) are used.

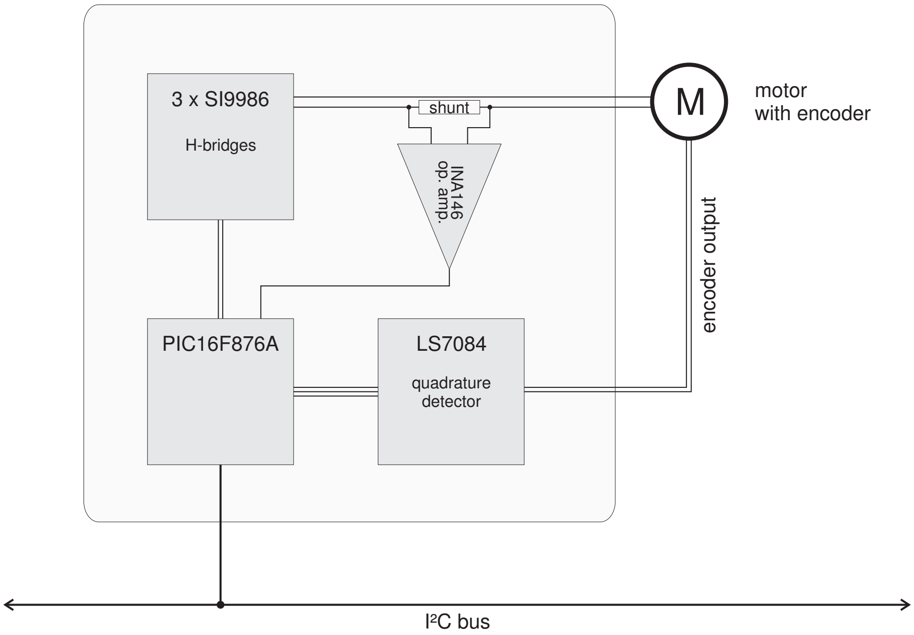


Fig. 9: Block schema of the PD controller of the body (and legs) elements

Each element contains three printed circuits (a power board, a PD motor controller and a small internal water detector) connected with a flat cable, a DC motor with an integrated incremental encoder, a set of gears (which uses two additional printed circuits as mechanical support) and a rechargeable Li-Ion battery. A view of an open element can be seen in Fig. 11. In opposition to the old elements, where all connections were realized by soldering the wires directly on the printed circuit boards, the new circuits use MicroMatch connectors for all the interconnections (bus, battery, motor and inter-circuit connection); only the water detector (which was added later to the design; see below) uses directly soldered wires for space reasons. The elements are completely independent from each other (both electrically and mechanically). The density of the robot elements is slightly lower than 1 kg/m^3 (the density of the old elements was slightly higher, therefore the first robot was not buoyant). The battery is placed at the bottom of the elements to have the center of mass below the geometrical center, therefore ensuring the vertical stability of the robot during both swimming and crawling.

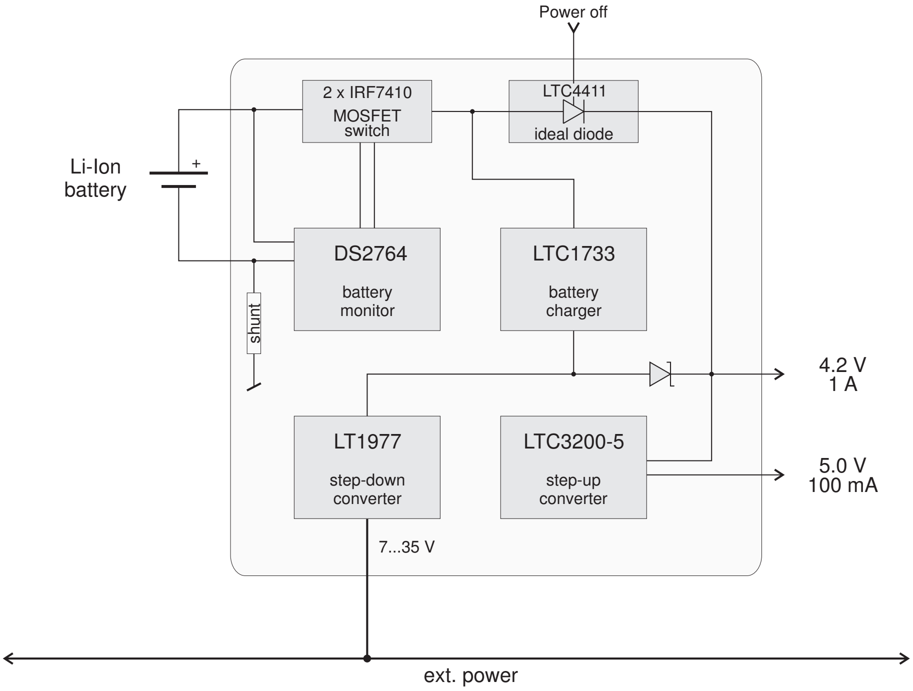


Fig. 10: Block schema of the power circuits of the body (and legs) elements

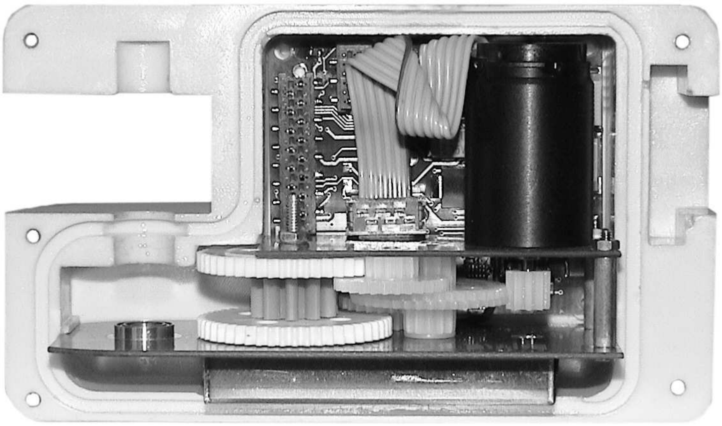
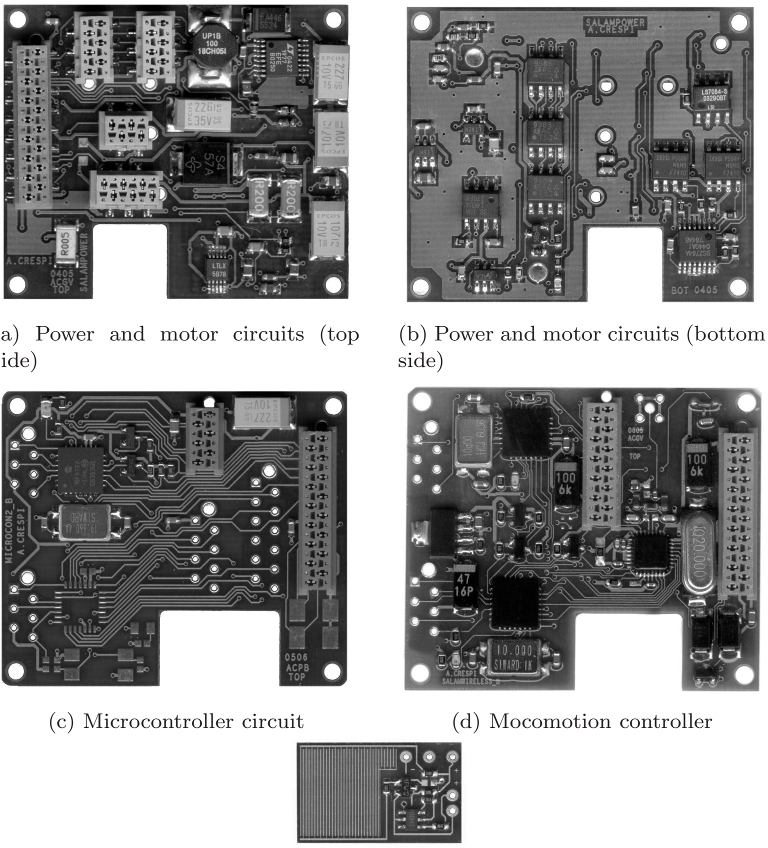


Fig. 11: Internal view of a body element (real size). The output axis is not mounted



(a) Power and motor circuits (top side)

(b) Power and motor circuits (bottom side)

(c) Microcontroller circuit

(d) Mocomotion controller

(e) Water sensor

Fig. 12: Pictures of the printed circuit boards of the robot (real size)

In this description, for simplicity, we will not distinguish on which of the printed circuits each component is located. The motor controller is based on a PIC16F876A microcontroller, and is basically the same of the first prototype. It is connected to the I^{2}C bus of the robot through a simple bidirectional repeater (built using two BSS138 MOS transistors), which is very useful to protect the microcontroller internal drivers. The motor has an integrated magnetic incremental encoder, which generates 512 pulses for every complete rotation of the motor axis. The encoder is connected to a LS7084 quadrature detector that filters and decodes the signals coming from the encoder, generating a direction flag and a clock signal, which are connected to the microcontroller. Compared to the first generation elements, the potentiometer has been removed to simplify the mechanical structure.

The motor coil is powered through three SI9986 buffered H-bridges connected in parallel (each of which has a maximum current of 1 A; the maximal current that can be drawn by the motor is thus 3 A). These H-bridges are driven by the microcontroller with a Pulse-Width Modulation (PWM) signal, allowing the speed of the motor to be changed by modifying the duty cycle of the control signal.

To measure the current used by the motor (and then, indirectly, its torque), a couple of 0.2 Ω resistors in parallel are inserted between the output of the H-bridges and the motor. The voltage drop obtained on these resistors is amplified by a INA146 operational amplifier and sent to an analog input of the microcontroller. The negative power (−5 V) for the operational amplifier is generated using a small capacitive inverter (MAX1719).

The power supply part of the electronics has been completely redesigned compared to the first prototype. A battery monitoring and protection circuit, which was missing, has also been included. The circuit generates the voltage required by most of the electronics (5 V) using a capacitive charge-pump step-up converter (LTC3200-5). All the electronics can be either powered by the internal Li-Ion battery, or by an external power source (connected to the last element and distributed internally to all elements). When no external power source is connected, the battery (connected to the rest of the circuit through a DS2764 battery monitoring/protection circuit that controls two IRF7410 power MOSFETs) directly powers the motor. When an external power source is connected, an inductive step-down converter (LT1977) generates a voltage of approximately 4.6 V, which can both replace the battery voltage (to power the motor and the step-up converter) and power the LTC1733 battery charger. The circuit accepts up to 35 V (to reduce as much as possible the current on the internal wires, which have a limited section). The switch between the internally generated 4.6 V and the battery is realized with a LTC4411 “ideal diode” and a SS34 Schottky diode. The used battery is the same that was used in the previous prototype and has a capacity of 600 mAh; it can power an element for approximately two hours of continuous use in normal conditions. When empty, the battery can be recharged in approximately one hour. The battery protection circuit disconnects the battery when its voltage drops below a critical threshold, thus preserving it from the often irreversible complete discharging. The circuit can also measure the instantaneous and accumulated current used by the circuit (or by the battery, during charging), and the battery voltage. This information could be read out using an I^{2}C bus, but these signals are currently left unconnected on the power card, to limit the total number of devices on the bus (which is global to the robot).

A signal coming from a reed contact placed in one of the elements allows the user to switch off the robot by placing a magnet on it. This solution was found to be simpler than using a big waterproof switch. This signal is connected to the enable pin of the aforementioned LTC4411 (no current is drawn, the signal can therefore be directly generated using one of the batteries).

The water detector circuit (fig. 12e), used internally to detect and localize any leakage, is placed at the bottom of the element. It has been introduced in the current elements to ease the detection of water leakages. It has a sensitive surface of about 1 cm^2, consisting of several parallel tracks, half of which are connected to the power source through a resistor. When water (or a big amount of moisture) is on this surface, it acts like a resistor between the power source and the base of an NPN transistor, which begins to conduce. When water is detected, the circuit blinks a LED fixed through the top of the element, therefore allowing the user to immediately detect the leakage and its position (i.e., the concerned element). The LED blinking is implemented using a PIC10F200 microcontroller, which in normal conditions accepts an incoming control signal for the LED and transparently replicates it on the output, hence permitting the LED to be used for other purposes. The introduction of this water detector dramatically simplified the handling of water leakages in the robot, which can now be localized without unmounting all the elements.

The 2.83 W DC motor (Faulhaber 1724 T 003 SR) has a maximum torque of 4.2 mN·m and drives a gearbox with a reduction factor of 125. It is approximately four times faster and stronger than the motor used in the previous generation of elements, therefore allowing higher amplitudes and oscillation frequencies to be reached. The output axis of the gears is fixed to the connection piece, which is inserted into the next element. Six wires are inserted into the axis, and connected to the power boards of two adjacent elements: two are used for the external power, two for the I^{2}C bus, one for the power switch and the last one is reserved for future usage and currently unconnected.

3.2 Limb elements

The limb elements have been designed mainly as legs for the salamander robot, but can indeed be used for other purposes (for example the pectoral fins of the BoxyBot fish robot [12]). Each limb element includes a pair of identical circuits (one for the left limb and one for the right one). The design is unlike a real animal limb: this element has an axis capable of continuous rotation as output (and thus only one degree of freedom), similarly to robots using whegs (i.e., wheel-legs, see [28, 32]). This gives to the element both flexibility (it can be used for other purposes than legs) and simplicity (only one motor and gearbox per limb).

These elements are based on the same electronics of the body elements, however, as the printed circuits are also used as mechanical support for the gears and the motor, the components are differently distributed between the circuits. Additionally, an infrared LED/phototransistor couple allows the de-

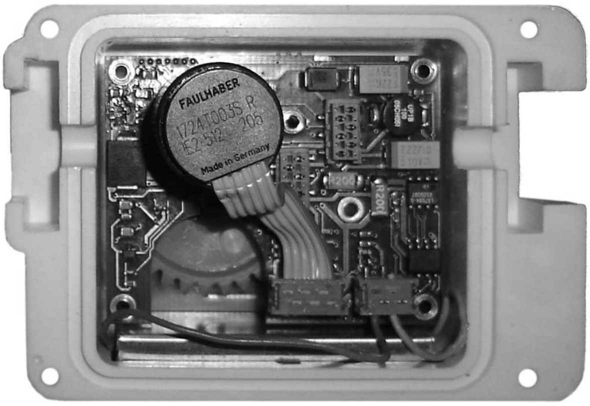


Fig. 13: Internal view of a limb element (real size)

tection of the absolute position of the output axis (using a hole in the last wheel of the gearbox), in order to automatically align it when powering up the robot.

3.3 Locomotion controller circuit

The locomotion controller circuit has been designed to meet the following criteria:

- To provide a simple but flexible locomotion controller with low energy consumption. It should be possible to implement on it any control algorithm, following the needs of the user (in the work presented here, CPG-based controllers have been implemented on it).
- To have bidirectional radio communication capabilities (both on ground and under water) for remote control and measure.

The circuit is placed inside an empty body element (i.e., without the motor and the gearbox); a variation of the same circuit without the radio communication functions has been used for controlling the BoxyBot fish robot [12]. A block schema of the controller electronics can be seen in Fig. 14. The circuit is based on a PIC18F2580 microcontroller, which is master on the I^{2}C bus of the robot. It can implement a locomotion controller (for example, a CPG) and sends out the setpoints to the motor controllers of each element in real time. The main microcontroller communicates, using a local serial line, with a PIC16LF876A microcontroller, which controls a nRF905 radio transceiver. The radio communication is handled by this separate microcontroller for sim-

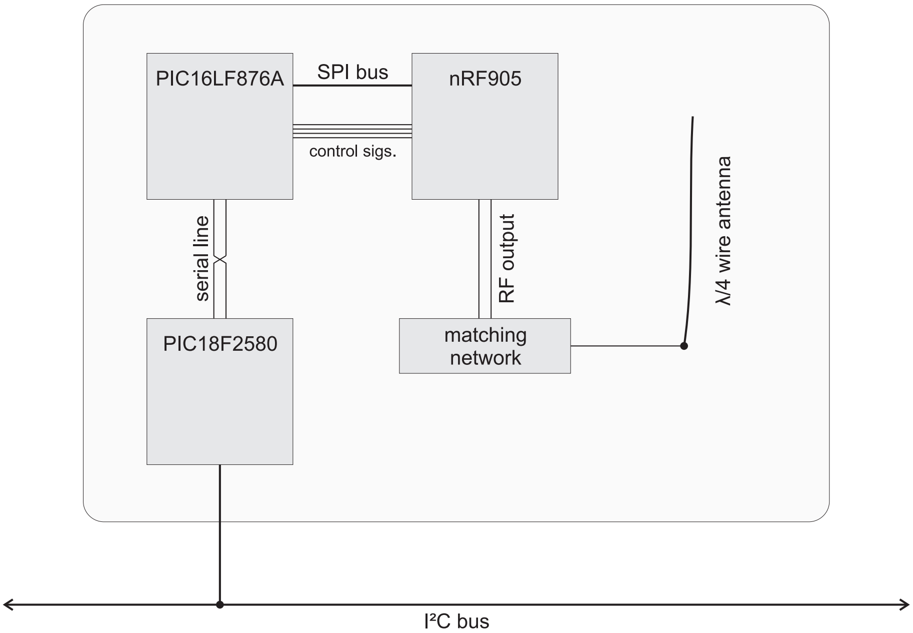


Fig. 14: Block schema of the electronics of the locomotion controller circuit

plicity, and because the PIC18F2580 can not handle hardware SPI and I^{2}C at the same time. The antenna is internal to the element and consists of a simple $\lambda/4$ wire (where λ is the wave length of the used frequency). The radio system uses the 868 MHz ISM band: preliminary experiments showed that a 10 mW signal (the power transmitted by the nRF905) on this frequency can penetrate in water up to at least 30 cm (the maximum tested depth). The more common 2.4 GHz band has not been used because it is heavily absorbed by the water. The maximal bandwidth is approximately 50 kbps, largely enough to send control commands and parameters to the online trajectory generator.

The software running on the locomotion controller can easily be reprogrammed with an external programming connector placed on the element.

4 Experiments

Several experiments have been done to characterize the locomotion of the robot. We first present the measures of how the locomotion speed is related to the input drive of the CPG, then compare how the robot locomotion is similar to the one of real salamanders in kinematic terms (focusing on lateral displacements of the body).

4.1 Speed as function of drive

The speed of the robot has been measured for 18 different drive values (between 1.0 and 3.0 for walking, and between 3.001 and 5.0 for swimming, with a step of 0.25). Each measure has been repeated 5 times, giving a total of 90 measures.

For walking, the speed has been measured by taking the time used to travel a given distance (i.e., 2 m). For swimming, the procedure was the same, but the distance was reduced to 1 m and the measure only started after an acceleration space of approximately 50 cm, to approach steady-state swimming as close as possible. The results are plotted in Fig. 15.

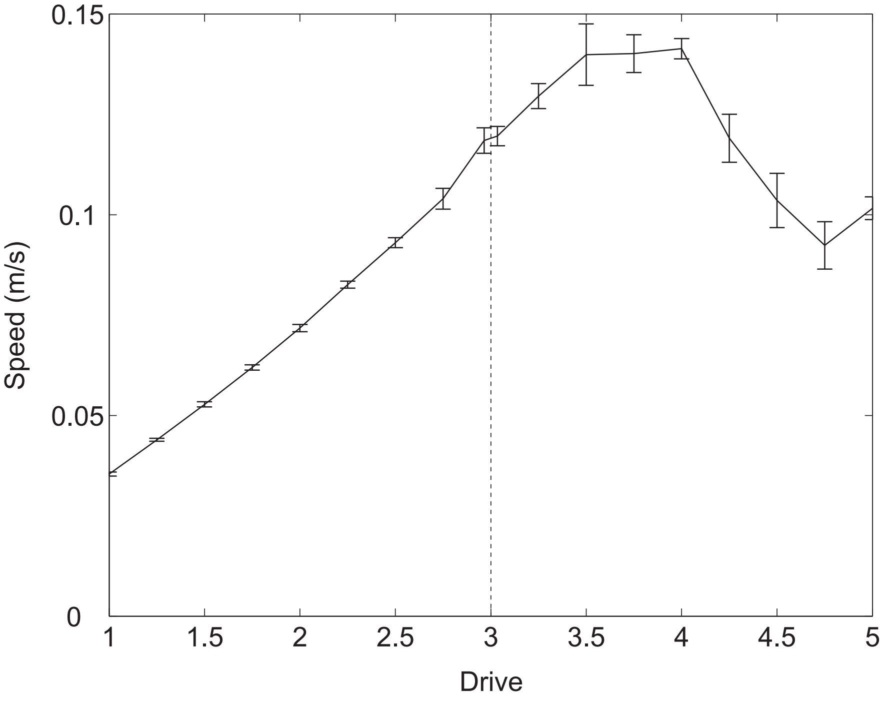


Fig. 15: Speed of the salamander robot for walking ($d \leq 3$) and swimming ($d > 3$)

For walking, the speed almost linearly increases over the whole range of drives, from $3.54 \cdot 10^{-2}$ m/s for $d = 1$ to $1.19 \cdot 10^{-1}$ m/s for $d = 3$. For swimming, the speed increases linearly from $1.20 \cdot 10^{-1}$ m/s for $d = 3.001$ to $1.40 \cdot 10^{-1}$ m/s for $d = 3.5$, then stabilizes around this value up to $d = 4$. For $d > 4$, the speed decreases, with a minimal value of $9.24 \cdot 10^{-2}$ m/s for $d = 4.5$. The decrease is mainly owing to the torque limits of the motors, which are not enough to follow the desired trajectories in water at high frequency, with the consequence that the speed saturates, and then for higher drives, the travelling wave begins to deform, with the resulting decrease of speed.

4.2 Kinematic measurements

To compare the movements of the robot with those of the real animal, kinematic measurements have been done on the robot, using a custom video tracking system. These data have been compared with kinematic recordings of *Pleurodeles waltlii* salamanders, which have been provided by Isabelle Delvolvé (INSERM, Bordeaux).

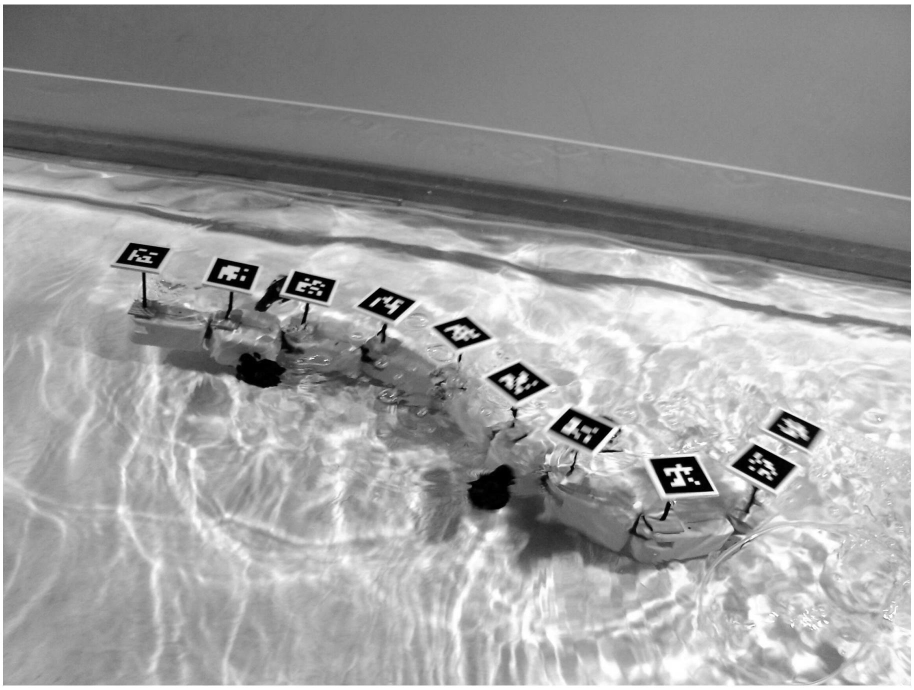


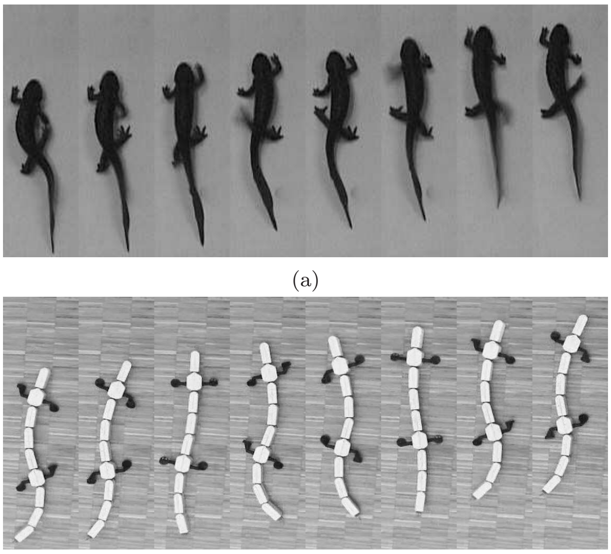
Fig. 16: The salamander robot swimming with the tracking markers fixed on it

The robot was filmed from above at 15 frames/s with a Basler A602fc-2 camera using an 8 mm C-mount lens. The frame data acquired over an IEEE1394 link was processed in real time with a custom program using the ARTag library [15] to extract the (x, y) coordinates of the markers (sort of 2D barcodes, see fig. 16) placed on the robot. For body elements, the markers were placed at the rotation center of the output axis; for head and limb elements, they were placed at the same distance from the element's border than on body elements. The coordinates have been exported in CSV files and then imported in MATLAB for processing and analysis, like for the salamander. The tracking markers had a size of 55 × 55 mm. To minimize motion blur, the exposure time of the camera was set around 2 ms, using two 500 W halogen projectors for lighting. For walking, they were fixed on the top of the robot with double-sided adhesive tape. For swimming, they were fixed on a PVC support having the same size of the marker and placed 75 mm above the robot (using a rigid PVC cylinder of diameter 4 mm), to ensure that the markers were always out of the water during tracking (Fig. 16). The measures were repeated five times for each drive level. For walking, the camera field of view was always containing two complete cycles; for swimming, this varied between two and five cycles.

For illustration, snapshots from videos (without the tracking markers) for one locomotion cycle for walking and swimming can be seen in Figs. 17 and 18, respectively.

A detailed comparison of the lateral displacements during a complete locomotion cycle can be found in Figs. 19 and 20. The two gaits are easily distinguished: walking uses standing waves, whereas swimming uses a travelling wave along the body. As it can be seen in the figures, the generated waves are similar for the salamander and the robot, although there are some small differences.

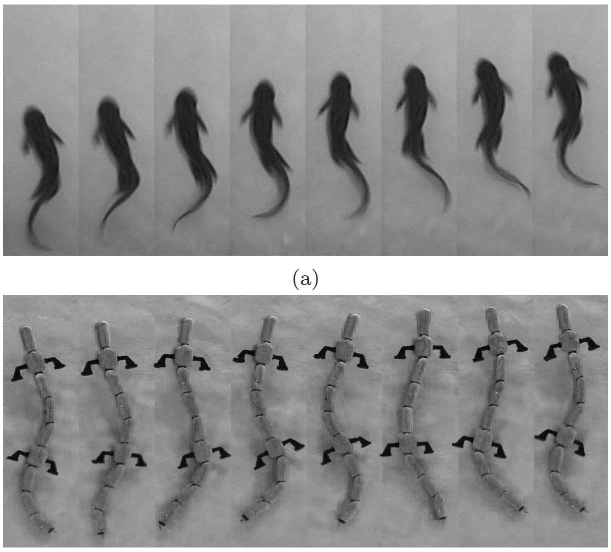
The envelopes of lateral displacement of each marker (relative to the direction of motion) measured with the video tracking are plotted in Figs. 21 (walking) and 22 (swimming), with the corresponding data of the animal. For walking, the motion is qualitatively similar for the robot and the animal; both have minimal lateral displacements at the girdles (which are not exactly at the same relative position). The main differences are the displacements of the queue (the salamander maintains the tip of its queue mostly straight, whereas the robot moves it) and of the head (the robot lacks a joint in the neck, therefore producing a greater lateral displacement of the head, compared to the real salamander that makes compensatory neck movements). For swimming, the motion is also similar for the robot and the salamander, but with more differences than for walking. Particularly, the lateral displacements of the robot are higher than those of the salamander between the girdles. This can be explained by the lack of hinge joints in the limb elements of the robot, and by their increased weight.



(a)

(b)

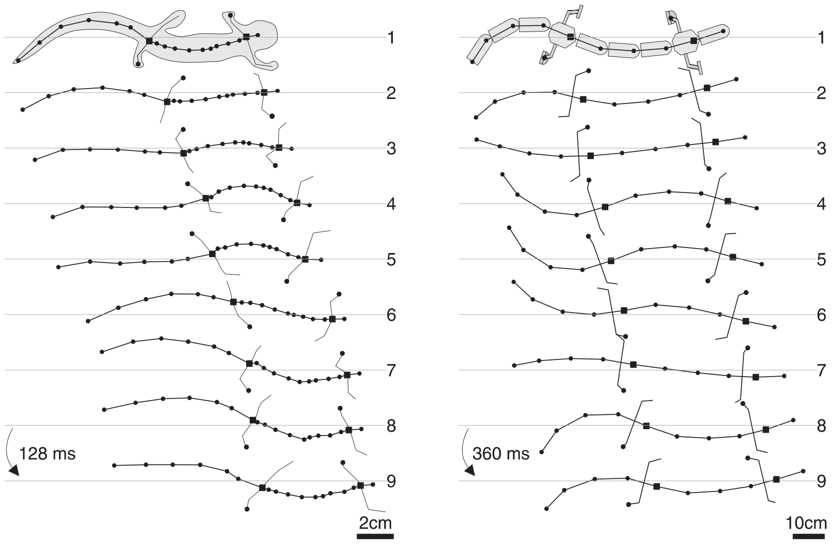
Fig. 17: Snapshots from videos for salamander (**a**) and robot (**b**) walking. The time step between the snapshots is 0.12 s for the salamander, and 0.20 s for the robot



(a)

(b)

Fig. 18: Snapshots from videos for salamander (**a**) and robot (**b**) swimming. The time step between the snapshots is 0.04 s for the salamander, and 0.12 s for the robot



(a) (b)

Fig. 19: Comparison of lateral displacements of the salamander (**a**) and robot (**b**) during walking. Velocities were 0.06 m/s (0.34 body lengths/s) for the animal, and 0.06 m/s (0.07 body lengths/s) for the robot ($d = 2.0$)

5 Future work

Although most of the limitations of the first generation of salamander robot were addressed and solved with the current generation, some problems remain partially unsolved, and new weaknesses (related to previously unavailable features) appeared.

Despite the efforts to correct problems with waterproofing, small water leakages still happen periodically. They mainly concern body elements, and owe to two main sources:

- The absence of a pair of screws (due to lack of space) at the horizontal center of the elements causes an insufficient compression of the O-ring. This problem is partially solved by using a silicone based sealant around the O-ring when closing the elements.
- The forces applied to the output axis during vertical movements (e.g., when lifting the robot, or when it has to overcome a small obstacle) can detach the brass axis from the flexible polyurethane rubber into which it is inserted, thus opening the way for possible water leakages.

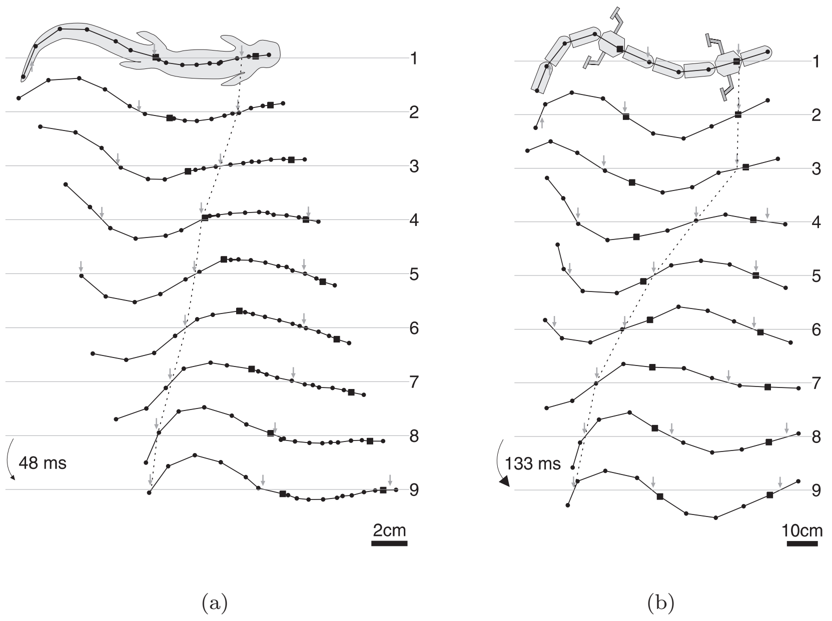


Fig. 20: Comparison of lateral displacements of the salamander (**a**) and robot (**b**) during swimming. Velocities were 0.17 m/s (0.89 body lengths/s) for the animal, and 0.11 m/s (0.13 body lengths/s) for the robot ($d = 4.0$)

These waterproofing problems have been addressed in a new third generation prototype (which is currently being tested as a snake robot, to which the legs will be added later), using a new closing system based on permanent magnets (instead of screws), thus also reducing the damages to the elements when unmounting them.

The limited computational capabilities of the PIC18F2580 used for the locomotion controller (a 8-bit microcontroller running at 40 MHz, obtaining a speed of 10 MIPS) required strong optimization of the code implementing the CPG locomotion controller. For instance, the CPG had to be implemented with fixed-point arithmetics and a large use of lookup tables, in order to obtain an acceptable execution speed (almost entirely using the available RAM and program memory). We are currently actively testing a new locomotion controller based on a faster microcontroller (a 60 MHz 32-bit ARM microcontroller).

The following improvements to the current generation of salamander robot have been applied in the third generation body elements now being tested:

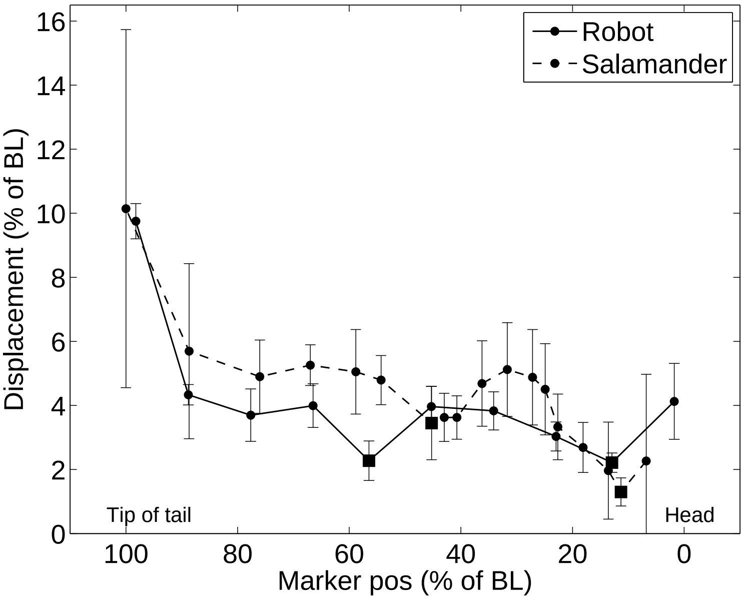


Fig. 21: Lateral displacements of the robot and real salamander during walking

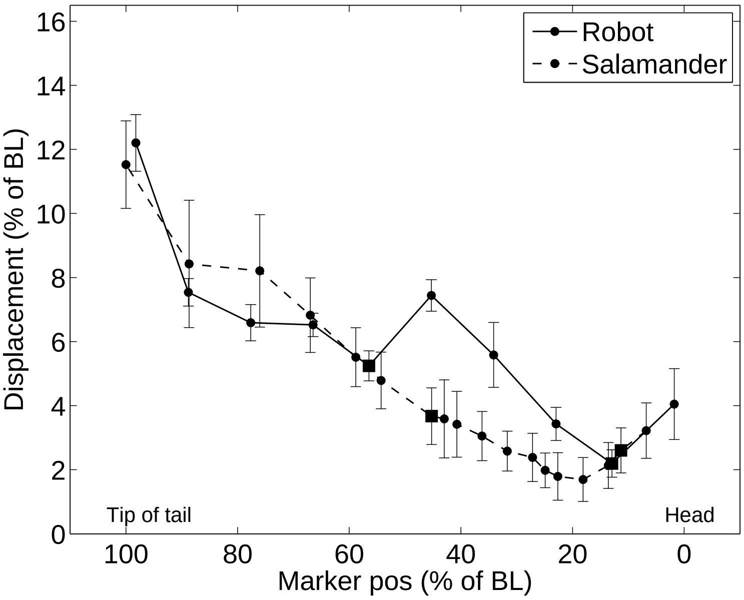


Fig. 22: Lateral displacements of the robot and real salamander during swimming

- The replacement of the current global I^{2}C bus with a faster and more reliable CAN bus. The I^{2}C is now used only inside elements for local communications between the different components.
- Adding the possibility to retrieve status information from the battery monitoring/protection circuit, therefore allowing the user to know the charging state of all the batteries, and to estimate the energy consumption of the robot and of each individual element.
- Integrating a new microcontroller (a PIC18F2580) in the elements, for interfacing to the CAN bus, and providing local computational capabilities for implementing a really distributed CPG.
- Having the possibility to remotely reprogram the trajectory generator program, using the radio link. The user has now the possibility to replace the program without using any special cable or programmer. This is especially important in the new robot, that does not feature an external programming connector on the head.
- Having more status information and diagnostic possibilities (both locally on the elements, using RGB LEDs, and remotely).
- Adding infrared distance/obstacle sensors on the head, to locally close the control loop and allow the robot to be autonomous in simple environments.

Further improvements to be implemented are the following:

- Implement a really distributed CPG, with each pair of oscillators placed in their own element, with couplings through the CAN bus. The new local microcontrollers and the use of a CAN bus will ease the implementation of this kind of controller.
- Adding a degree of freedom to the limb elements, which are currently rigid and interrupt the wave along the chain of elements; this will result in a better locomotion.
- Potentially adding a supplementary degree of freedom to have the possibility to lift the body (i.e., to remove the current limitation to planar locomotion).
- Adding other sensors (e.g., accelerometers or light sensors) and vision capabilities.

6 Conclusion

Realization of an amphibious salamander robot

Amphibious robotics is a rather challenging topic: the complete waterproofing requirements of amphibious robots, for instance, completely influence all aspects in the design process, rendering it more complex comparing to dry-land robots. However, the advantages of amphibious robots are evident: an amphibious robot like a salamander can deal with difficult environments which include water in any form (for example rain, partially flooded terrains, mud, etc.). This is a clear advantage, for instance, for outdoor robots used for inspection or exploration tasks. In our salamander robot, we used a modular approach that uses only two different types of elements (i.e., body and leg elements). The same modules can be used to construct other types of robots, some of which have been realized [11, 12].

Central pattern generators in robots

Central pattern generators are more and more used for the control of robots. This biologically inspired control technique is well suited for the control of complex robots having multiple degrees of freedom, as they can generate coordinated control signals for all the joints, receiving only simple low-dimensional inputs. This means that CPGs are a good control method for implementing interfaces to be used by human operators for the interactive control of such robots. They also provide a sort of "abstraction layer" that can hide the complexity of the robot to the end user, also rendering it possible to control different types of robots with the same set of control signals. Finally, CPGs can also easily deal with continuously changing input signals possibly featuring discontinuities, like those that can be provided by a human operator or learning algorithm. In this work, a CPG model for controlling a salamander robot, inspired from the spinal cord structure of real salamanders, has been developed and successfully implemented to run on the onboard locomotion controller. Two drive signals are sufficient to modulate speed, direction, and type of gait; this also simplifies the control by a human user or higher level controller. One of the advantages of the use of CPGs is the possibility to easily include sensory feedback in the control loop, but this has still to be implemented in the salamander robot (as the current generation does not include any sensor, except for the motor incremental encoder).

Contributions to biology

Finally, this robot proved to be a useful tool for verifying biological hypotheses. As explained in Section 1.1, robots could provide for example a "body" with which models of locomotion controllers can be tested to verify whether they actually generate locomotion in a real environment. We presented a model of central pattern generator that explains the locomotion control in salamanders and the transition between walking a swimming, using a single control parameter. This model has been successfully implemented and tested on the salamander robot, demonstrating that it can actually generate forward motion of a body, using real actuators, in a real environment.

References

1. Ayers, J., Crisman, J.: The lobster as a model for an omnidirectional robotic ambulation control architecture. In: R. Beer, R. Ritzmann, T. McKenna (eds.) Biological neural networks in invertebrate neuroethology and robotics, pp. 287–316. Academic Press (1993)
2. Breithaupt, R., Dahnke, J., Zahedi, K., Hertzberg, J., Pasemann, F.: Robo-salamander – an approach for the benefit of both robotics and biology. In: Proceedings of the 5th International Conference on Climbing and Walking Robots (CLAWAR 2002) (2002)
3. Cabelguen, J., Bourcier-Lucas, C., Dubuc, R.: Bimodal locomotion elicited by electrical stimulation of the midbrain in the salamander Notophthalmus viridesecens. The Journal of Neuroscience **23**(6), 2434–2439 (2003)
4. Chirikjian, G., Burdick, J.: Design, implementation, and experiments with a thirty-degree-of-freedom 'hyper-redundant' robot. In: ISRAM 1992 (1992)
5. Choi, H., Ryew, S.: Robotic system with active steering capability for internal inspection of urban gas pipelines. Mechatronics **12**, 713–736 (2002)
6. Cohen, A.: Evolution of the vertebrate central pattern generator for locomotion. In: A.H. Cohen, S. Rossignol, S. Grillner (eds.) Neural Control of Rhythmic Movements in Vertebrates. Jon Wiley & Sons (1988)
7. Conradt, J., Varshavskaya, P.: Distributed central pattern generator control for a serpentine robot. In: ICANN 2003 (2003)
8. Crespi, A., Badertscher, A., Guignard, A., Ijspeert, A.: An amphibious robot capable of snake and lamprey-like locomotion. In: Proceedings of the 35th international symposium on robotics (ISR 2004) (2004)
9. Crespi, A., Badertscher, A., Guignard, A., Ijspeert, A.: AmphiBot I: An amphibious snake-like robot. Robotics and Autonomous Systems **50**(4), 163–175 (2005)
10. Crespi, A., Badertscher, A., Guignard, A., Ijspeert, A.: Swimming and crawling with an amphibious snake robot. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005), pp. 3035–3039 (2005)
11. Crespi, A., Ijspeert, A.: AmphiBot II: An amphibious snake robot that crawls and swims using a central pattern generator. In: Proceedings of the 9th International Conference on Climbing and Walking Robots (CLAWAR 2006) (2006)
12. Crespi, A., Lachat, D., Pasquier, A., Ijspeert, A.: Controlling swimming and crawling in a fish robot using a central pattern generator. Autonomous Robots **25**, 3–13 (2008)
13. Delvolvé, I., Bem, T., Cabelguen, J.M.: Epaxial and limb muscle activity during swimming and terrestrial stepping in the adult newt, *Pleurodeles waltl*. Journal of Neurophysiology **78**, 638–650 (1997)
14. Dowling, K.: Limbless locomotion: Learning to crawl with a snake robot. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1997)
15. Fiala, M.: ARTag revision 1. A fiducial marker system using digital techniques. Tech. Rep. NRC 47419, National research council Canada, institute for information technology (2004)
16. Frolich, L., Biewener, A.: Kinematic and electromyographic analysis of the functional role of the body axis during terrestrial and aquatic locomotion in the salamander *Ambystoma tigrinum*. Journal of Experimental Biology **62**, 107–130 (1992)
17. Fukuoka, Y., Kimura, H., Cohen, A.: Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. The International Journal of Robotics Research **22**(3–4), 187–202 (2003)
18. Gao, K.Q., Shubin, N.: Late jurassic salamanders from northern China. Nature **410**, 574–577 (2001)
19. Hiraoka, A., Kimura, H.: A development of a salamander robot - design of a coupled neuro-musculoskeletal system. In: Proceedings of the Annual Conference of the Robotics Society of Japan, Osaka (2002). (Paper in Japanese)

20. Hirose, S.: Biologically Inspired Robots (Snake-like Locomotors and Manipulators). Oxford University Press (1993)
21. Hirose, S., Fukushima, E.: 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, pp. 48–63. Springer-Verlag (2002)
22. Ijspeert, A., Crespi, A., Ryczko, D., Cabelguen, J.M.: From swimming to walking with a salamander robot driven by a spinal cord model. Science **315**, 1416–1420 (2007)
23. Klaassen, B., Paap, K.: GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control. In: Proceedings of 1999 IEEE International Conference on Robotics and Automation (ICRA 1999), pp. 3014–3019. IEEE (1999)
24. Lee, T., Ohm, T., Hayati, S.: A highly redundant robot system for inspection. In: Proceedings of the conference on intelligent robotics in the field, factory, service, and space (CIRFFSS '94), pp. 142–149. Houston, Texas (1994)
25. McIsaac, K., Ostrowski, J.: A geometric approach to anguilliform locomotion: Simulation and experiments with an underwater eel-robot. In: Proceedings of 1999 IEEE International Conference on Robotics and Automation (ICRA 1999), pp. 2843–2848. IEEE (1999)
26. Miller, G.: Neurotechnology for biomimetic robots, chap. Snake robots for search and rescue. Bradford/MIT Press, Cambridge London (2002)
27. Paap, K., Dehlwisch, M., Klaassen, B.: GMD-snake: a semi-autonomous snake-like robot. In: Distributed Autonomous Robotic Systems 2. Springer-Verlag (1996)
28. Quinn, R.D., Nelson, G., Bachmann, R., Kingsley, D., Offi, J., Ritzmann, R.: Insect designs for improved robot mobility. In: Proceedings of the 4th International Conference on Climbing and Walking Robots (CLAWAR 2001) (2001)
29. Roth, G., Nishikawa, K., Naujoks-Manteuffel, C., Schmidt, A., Wake, D.B.: Paedomorphosis and simplification in the nervous system of salamanders. Brain, Behavior and Evolution **42**, 137–170 (1993)
30. Roth, G., Nishikawa, K., Wake, D.B.: Genome size, secondary simplification, and the evolution of the brain in salamanders. Brain, Behavior and Evolution **50**, 50–59 (1997)
31. Saito, M., Fukaya, M., Iwasaki, T.: Serpentine locomotion with robotic snakes. IEEE Control Systems Magazine **22**, 64–81 (2002)
32. Saranli, U., Buehler, M., Koditschek, D.: RHex – a simple and highly mobile hexapod robot. The International Journal of Robotics Research **20**(7), 616–631 (2001)
33. Stefanini, C., Orlandi, G., Menciassi, A., Ravier, Y., La Spina, G., Grillner, S., Dario, P.: A mechanism for biomimetic actuation in lamprey-like robots. In: Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2006), pp. 579–584 (2006)
34. Takayama, T., Hirose, S.: 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 (2001)
35. Takayama, T., Hirose, S.: Amphibious 3D active cord mechanism "HELIX" with helical swimming motion. In: Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002), pp. 775–780. IEEE (2002)
36. Umetani, Y., Hirose, S.: Biomechanical study of active cord mechanism with tactile sensors. In: Proceedings of the 6th international symposium on industrial robots, pp. c1-1–c1-10. Nottingham (1976)
37. Webb, B.: What does robotics offer animal behaviour? Animal Behaviour **60**, 545–558 (2000)
38. Webb, B.: Can robots make good models of biological behaviour? Behavioral and brain sciences **24**, 1033–1050 (2001)
39. Webb, B.: Robots in invertebrate neuroscience. Nature **417**, 359–363 (2002)
40. Webb, B., Reeve, R.: Reafferent or redundant: Integration of phonotaxis and optomotor behavior in crickets and robots. Adaptive Behavior **11**(3), 137–158 (2003)

41. Wilbur, C., Vorus, W., Cao, Y., Currie, S.: Neurotechnology for biomimetic robots, chap. A Lamprey-Based Undulatory Vehicle. Bradford/MIT Press, Cambridge London (2002)
42. Worst, R.: Robotic snakes. In: Third German Workshop on Artifical Life, pp. 113–126. Verlag Harri Deutsch (1998)
43. Yamada, H., Chigisaki, S., Mori, M., Takita, K., Ogami, K., Hirose, S.: Development of amphibious snake-like robot ACM-R5. In: Proceedings of the 36^{th} International Symposium on Robotics (2005)