AmphiBot II: An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator

Alessandro Crespi, Auke Jan Ijspeert
Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
Station 14, CH-1015 Lausanne, Switzerland
email: {alessandro.crespi, auke.ijspeert}@epfl.ch

Abstract— **This article presents AmphiBot II, an amphibious snake robot designed for both serpentine locomotion (crawling) and swimming. It is controlled by an on-board central pattern generator (CPG) inspired by those found in vertebrates. The CPG is modelled as a chain of coupled nonlinear oscillators, and is designed to produce travelling waves. Its parameters can be modified on the fly. We present the hardware of the robot and the structure of the CPG, then the systematic parameter tests done in simulation and with the real robot to characterize how the speed of locomotion depends on the parameters determining the frequency, amplitude and wavelength of the body undulation.**

Keywords— **AmphiBot II, amphibious robot, snake robot, central pattern generator, swimming, crawling**

I. Introduction

Snake-like robots have been studied for many years, particularly for their ability to deal with difficult environments, in which other types of robots often fail. One of first snake-like robot, named an *active cord mechanism* [1], has been constructed in 1972. Since then a variety of different snake robots have been designed, see [2], [3], [4], [5], [6], some of which are currently used for the inspection of pipes [7], for example. A review of snake robots can be found in [8] and [9]. Most of these robots have been designed for locomotion on ground, and only a few working examples of swimming snake robots currently exist. The most interesting ones are the eel robot *REEL II* [10], the lamprey robot built at Northeastern University [11] and the spirochete-like HELIX-I [12].

The goals of this project are two-fold: (1) to build an amphibious snake-like robot that can both crawl and swim for outdoor robotics tasks, taking inspiration from snakes and elongate fishes such as lampreys, and (2) to demonstrate the use of central pattern generators (CPGs) as a powerful method for online trajectory generation for crawling and swimming in a real robot. To the best of our knowledge, AmphiBot II is one of the first amphibious snake robots controlled online by a central pattern generator. As will be presented in this article, CPGs can be implemented as systems of coupled nonlinear oscillators which produce coordinated rhythmic patterns (in this case travelling waves). These patterns can be designed to be stable, i.e. to exhibit limit cycle behavior, which makes them robust against external perturbations. In addition, they can easily be modulated online by the change of a few control parameters. This makes them ideally suited for online trajectory generation with a human in the loop: the user can modify CPG parameters to adjust the speed and direction of locomotion without dealing with the complexity of individually controlling each degree of freedom.

Fig. 1. The AmphiBot II robot.

The robot presented in this paper, AmphiBot II, is the new version of AmphiBot I [13], [14]. Compared to its predecessor, it features a significant number of improvements:

- A better mechanical design: the construction of the robot is greatly simplified, and all pieces can be assembled without soldering.
- More powerful motors: the maximal torque has been increased by a factor of 3.5.
- Wireless communication capabilities: the robot has now an internal transceiver, which can be used to control it remotely without any wires.
- Improved waterproofing.
- Onboard CPG: the motor commands are now generated online, directly in the robot, by a central pattern generator running on a microcontroller, therefore removing the need of running the controller on an external computer.

In the rest of the article, we will first describe the hardware of the robot (Section II). We then present our Central Pattern Generator model (Section III) and a simulator of the robot that we used for initial characterization (Section IV). Systematic characterizations with the simulated and real robots of the speed of locomotion as a function of different control parameters of the CPG are presented in Section V. We finish the article with two short sections on future work and conclusions.

II. Hardware description

The AmphiBot II robot is designed to be modular: it is constructed out of several identical segments, named elements. The robot described in this paper has 7 actuated elements (i.e. 7 degrees of freedom) and a head (which is externally identical to the other elements); however with the current electronics a robot with up to 127 segments can theoretically be built by simply adding other elements to the chain. The external cas-

ing of each element consists of two symmetrical parts, which are fixed together with screws. The elements are connected (both mechanically and electrically) using a compliant connection piece fixed to the output axis, which contains 6 wires. All parts of the robot body are molded using polyurethane resin lighted with glass microballs; the connection pieces are also molded with polyurethane. All the output axes of the elements are aligned, therefore producing planar locomotion. To ensure the waterproofing of the robot, a custom O-ring is used. An element has an external length of 9.4 cm, and a section of 5.5 by 3.7 cm. The total length of the robot presented in this paper is 77.2 cm, considering that the connection piece introduces a distance of 0.25 cm between two adjacent elements. The asymmetric friction with the ground, required for the robot to correctly crawl on the ground, is obtained by fixing a couple of passive wheels to each element with double-face adhesive tape. Currently the wheels are removed for swimming. The density of the robot is slightly lower than 1 kg/m^3, so that the robot floats under the surface when in water. The battery is placed at the bottom of the elements to have the center of mass below the vertical center, therefore ensuring the vertical stability of the robot during both swimming and crawling. During swimming, the robot is connected to a small aquarium pump through a highly flexible PVC tube: maintaining a little overpressure inside the elements avoids any possible leakage.

A. Actuated elements

Each element contains three printed circuits (a power board, a PD motor controller and a small 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. The elements are thus completely independent from each other (both electrically and mechanically). In this description, for simplicity, we will not distinguish on which of the printed circuits each component is located.

The power circuit generates the voltage required by most of the electronics (5 V) using a capacitive charge-pump step-up converter (LTC3200-5).

The motor controller is based on a PIC16F876A microcontroller, which runs a PD motor controller developed at the Autonomous Systems Laboratory of the EPFL. 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.

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.

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 30 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 has a capacity of 600 mAh, and 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 drop below a critical threshold, thus preserving it from the often irreversible complete discharging.

A signal coming from a reed contact (currently placed in the head element) allows 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, used internally to detect and localize any leakage, is placed at the bottom of the element. 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.

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. 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^2C bus, one for the power switch and the last one is reserved for future usage and currently unconnected.

B. Head element

The head element, like the body elements, has three printed circuits (a power board without all the motor-related circuits, a controller board, and a water detector).

The controller circuit is based on a PIC18F2580 microcontroller, which is master on the I^2C bus of the robot. It implements the CPG (described in section III), and sends out the setpoints to the motor controllers of each element in realtime. The main microcontroller communicates, using a serial line, with a PIC16LF876A microcontroller, which controls a nRF905 radio transceiver. The antenna is internal to the element and consists of a simple $\lambda/4$ wire. The radio system uses the 868 MHz ISM band: preliminar experiments showed that a 10 mW signal (the power transmitted by the nRF905) on this frequency can penetrate in water up to 30 cm (the maximum tested depth). The more common 2.4 GHz band has not been chosen because it corresponds to the resonant frequency of water and is therefore too much absorbed. The maximal bandwidth is approximately 50 kbps, largely enough to send control commands and parameters to the online trajectory generator.

III. Central Pattern Generator

The locomotion of vertebrates is controlled by central pattern generators (CPGs), which are neural networks producing coordinated oscillatory signals without oscillatory inputs [15]. They are located in the spinal cord and distributed in multiple oscillatory centers.

A central pattern generator is an elegant solution to the problem of online generating motor trajectories for hyper-redundant robots, like AmphiBot II: it takes simple input parameters (in this case oscillation amplitude, frequency and phase difference) and generates the oscillatory output signals, smoothing any discontinuities in the input. A simple sine-based controller, which can also be used to control a snake robot, will react instantaneously to any change of the input, thus sending abrupt changes to the motors.

Our CPG model is based on a system of amplitude-controlled phase oscillators, which is integrated by the microcontroller of the head (see section II-B) using the Euler method, with a time step of 10 ms and using fixed point arithmetics. It is a simpler version of the CPG presented in [16]. The CPG is a double chain of oscillators (see figure 2, where all the connections between the oscillators are visible). 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 and serpentine locomotion on ground. All the parameters of the CPG (amplitude, frequency, total wavelength) can be remotely set, thus allowing the locomotion of the robot to be completely controlled without any tether.

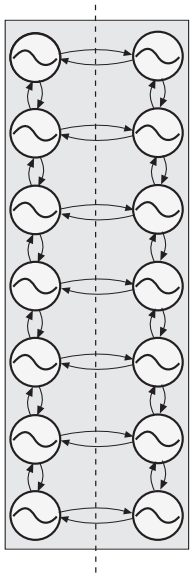


Fig. 2. Structure of the CPG used in the robot.

The CPG is implemented as the following system of coupled oscillators:

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

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}.

In this article, we chose the frequency and amplitude parameters to be equal for all oscillators, i.e. $\nu_i = \nu$ and $R_i = A/2$ for all i. 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 $\Delta\phi$ for the descending connections and to $-\Delta\phi$ for the ascending connections. We used $w_{ij} = 1$ for all connections and $a_i = 100$ for all oscillators.

With these settings, the CPG asymptotically converges to a limit cycle that is defined by the following equation for the i^{th} oscillator:

$$x_i(t) = \frac{A}{2} \cdot \Big(1 + \cos(2\pi\nu \cdot t + i\Delta\phi + \phi_0)\Big) \quad (2)$$

where ϕ_0 depends on the initial conditions of the system. This means that the system always stabilizes into a travelling wave whose frequency, amplitude, and

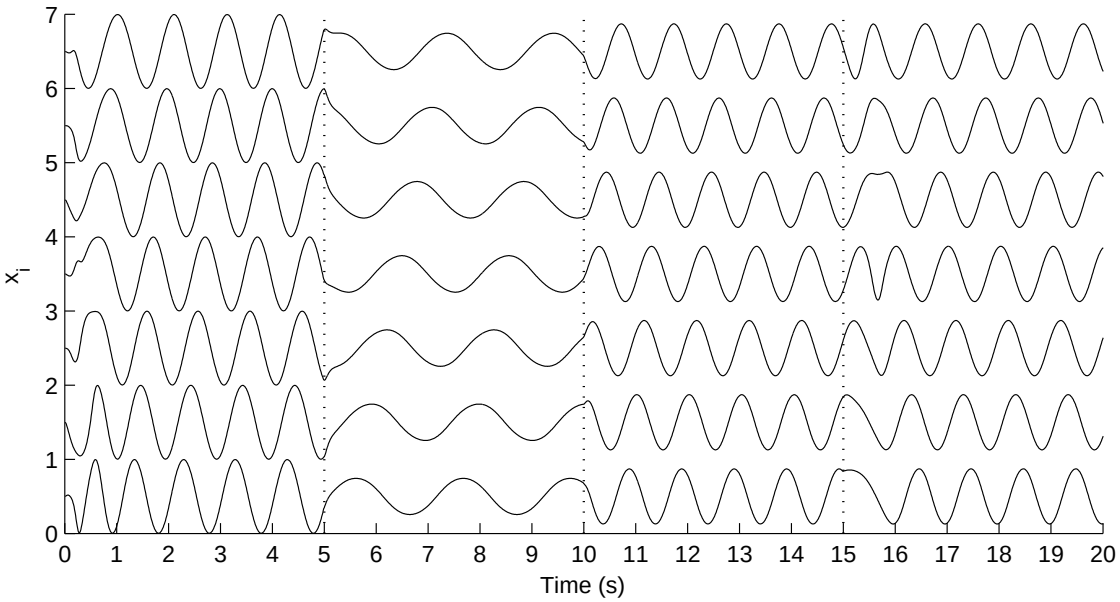


Fig. 3. Effect of changing the parameters of the CPG. Initial parameters are $R = 1$, $\nu = 1$ Hz and $N \cdot \Delta\phi = 1$. At $t = 5$ s, $R = 0.5$ and $\nu = 0.5$ Hz, at $t = 10$ s, $R = 0.75$ and $\nu = 1$ Hz. At $t = 15$ s, $N \cdot \Delta\phi = -1$.

phase lag are directly determined by the control parameters ν, A and $\Delta\phi$. These parameters can be modified online by a human user from a control PC using the wireless connection. The CPG will rapidly adapt to any parameter change and converge to the modified travelling wave after a short transient period. An example of how the CPG reacts to parameter changes can be observed in figure 3: when the parameters are changed, the oscillator smoothly converges to the new limit cycle, without any discontinuities in the outputs.

The setpoints sent to the PD motor controllers are the differences between the x_i signals of the oscillators on the left and right sides of the chain. This allows one to easily induce turning movements by setting different R values for the left and right oscillators. The oscillators on the side with the larger R value will oscillate with larger amplitudes and this therefore leads to an offset of the setpoints oscillations, i.e. the robot will tend to bend more to that side. The turning behaviours have not been studied in this article.

IV. Simulation

A model of the robot has been created with Webots [17]. It is controlled by the same CPG of the real robot (with the exception that it is implemented on a PC using standard floating point arithmetics) and has the same mechanical and physical properties of the real robot. The wheels are modelled with asymmetric friction (simulated with a simplification of the Coulomb friction model), using the friction coefficients $\mu_1 = 0.5$ (friction perpendicular to the speed) and $\mu_2 = 0.025$ (friction parallel to the speed).

For swimming, the hydrodynamic forces applied to each element of the robot are the Archimedes' force $\overrightarrow{F_A}$ and the drag forces $\overrightarrow{F_D}$:

$$\begin{aligned} \overrightarrow{F_A} &= -V_{el} \cdot \rho_{water} \cdot \vec{g} \\ F_{D_d} &= \tfrac{1}{2} \cdot \rho_{water} \cdot S_d \cdot C_d \cdot v_d^2 \ (d \in \{x, y, z\}) \end{aligned} \tag{3}$$

where V_{el} is the volume of the immersed part of the element, ρ_{water} is the density of the water, $\vec{g}$ is the gravity acceleration, S_d is the surface of the element's section perpendicular to the d axis, C_d is the drag coefficient relative to the d axis, and v_d is the speed of the element along the d axis. Although this model is relatively simple (e.g. it does not take into account the turbulence generated in the water by the movement of the robot), it is enough for a qualitative analysis of the swimming motion.

V. Locomotion characterization

Serpentine locomotion and anguilliform swimming require a travelling wave to be propagated from the head to the tail of the robot. In this section, we analyze the results of our experiments, where we systematically tested how the three control parameters of the CPG (amplitude A, phase difference $\Delta\phi$ and frequency ν) affect the locomotion speed of the robot, both on ground and in water. The parameters have been kept into a reasonable range: the amplitude between $\pm 10°$ and $\pm 60°$ (with a step of $10°$), the frequency between 0.2 Hz and 1.0 Hz (with a step of 0.2 Hz) and the phase difference between $0.25/N$ and $1.5/N$ (with a step of $0.25/N$). For the lowest phase ($0.25/N$), the amplitude has been limited to $\pm 40°$, as a greater amplitude would lead to collisions between the elements of the robot at some points of the cycle.

A. Simulation

The results of the simulated systematic tests at $\nu =$ 1.0 Hz (this frequency corresponds to the observed max-

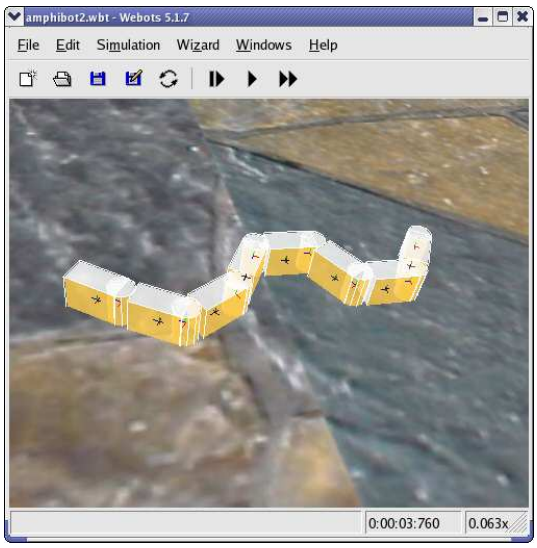


Fig. 4. A screenshot of the crawling simulation running in Webots.

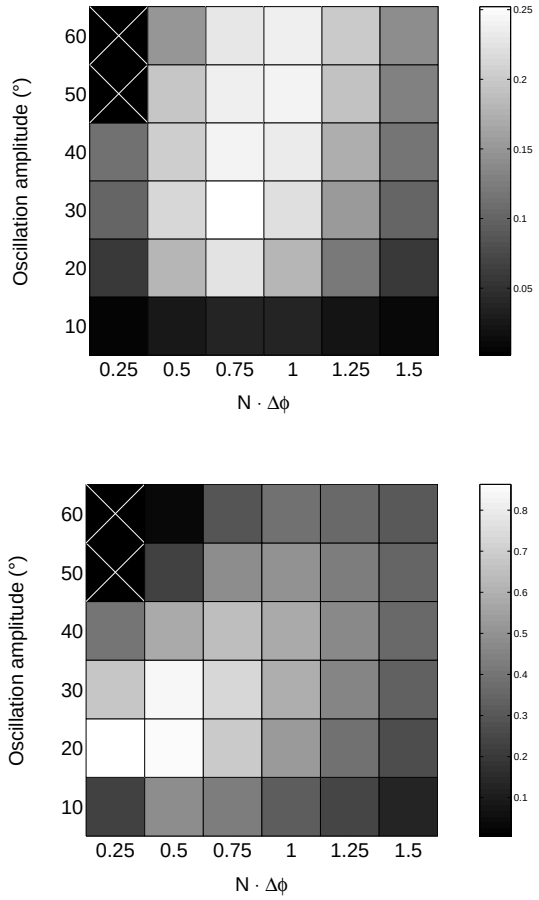


Fig. 5. Speeds (m/s) for the simulated systematic tests at $\nu = 1.0$ Hz (top: crawling; bottom: swimming).

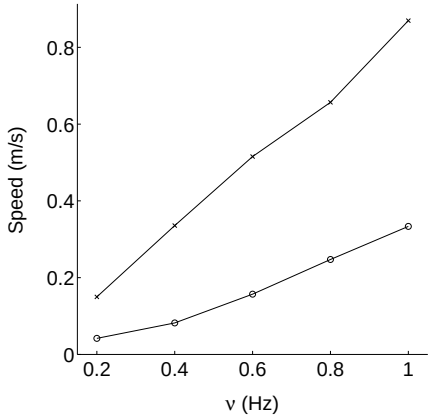


Fig. 6. Effect of the frequency on the simulated speeds (∘: crawling at $A = \pm 40°$ and $N \cdot \Delta\phi = 0.75$; ×: swimming at $A = \pm 20°$ and $N \cdot \Delta\phi = 0.25$).

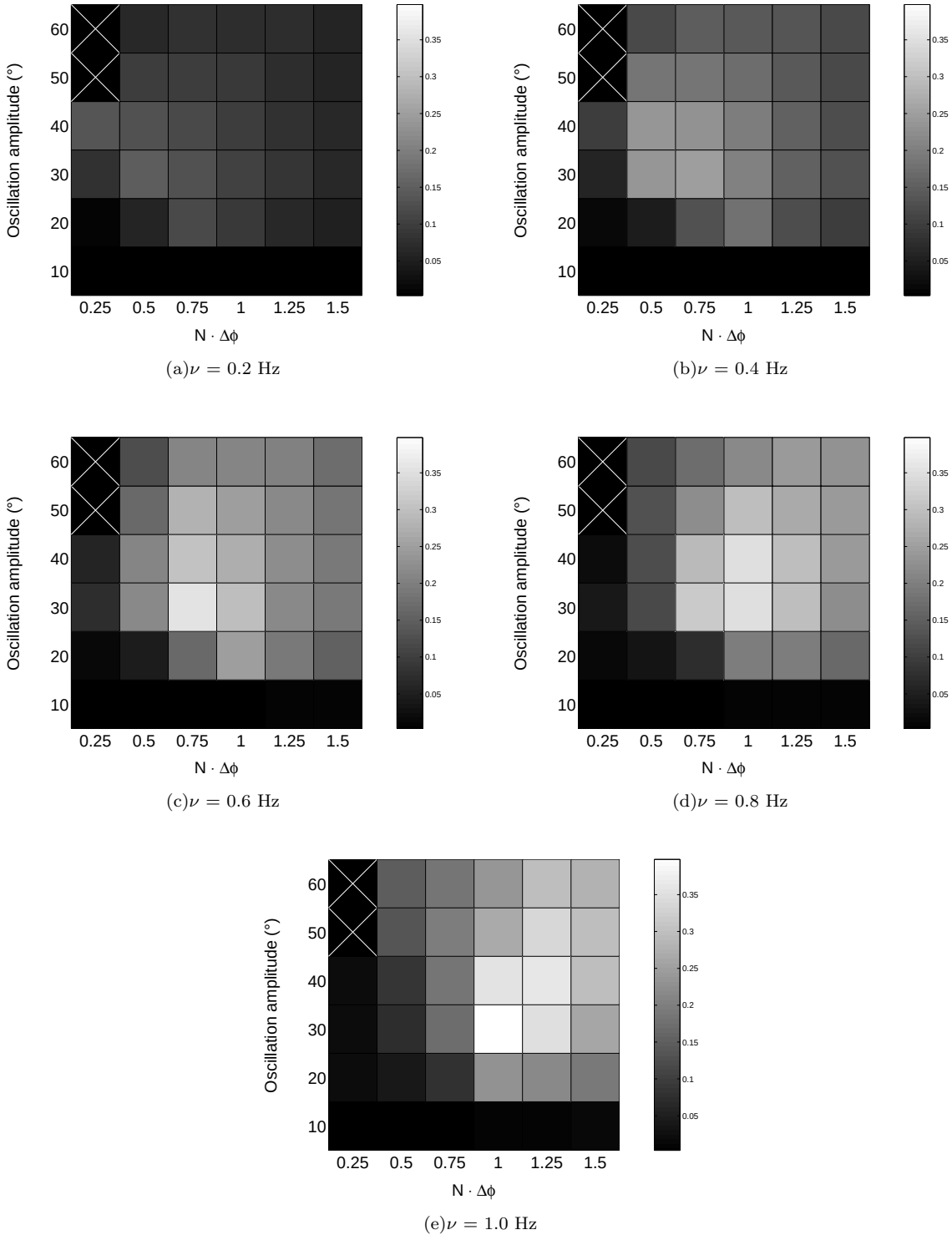
imal speed for both modes) are plotted in figure 5. The maximal speed obtained for crawling is 0.33 m/s, with $N \cdot \Delta\phi = 0.75$ and $A = \pm 40°$. For swimming, the maximal locomotion speed is 0.87 m/s, with $N \cdot \Delta\phi = 0.25$ and $A = \pm 20°$. For both modes, the optimum is relatively peaked (meaning that a small distance from the optimum causes a significant drop of the speed), without any local speed maxima. We systematically tested locomotion at other frequencies and found that the optimal amplitude and phase difference values for a given frequency did not change compared to the tests with $\nu = 1.0$ Hz (data not shown). The frequency seems to have a direct, almost linear, effect on the locomotion speed (see figure 6), without influencing the position of the optimum in terms of amplitude and phase difference both for swimming and crawling.

B. Crawling with the real robot

For each parameter set, the speed has been measured as follows: the robot was placed on a wooden floor, in front of an horizontal line, then started, and was stopped manually when reaching another line, parallel to the first one, at a distance of 2.00 m. The exact time between the start and stop commands was measured by the controlling PC. If the locomotion was visibly bad (i.e. if the first line was not reached after approximately 40 s), the robot was stopped before reaching the end line and the distance manually measured.

Depending on the parameters, the locomotion speed varied between 0 and 0.40 m/s (0.52 body lengths/s). In the tested range, the speed clearly increased with the increase of the frequency, and the optimum was always obtained with an amplitude of $\pm 30°$. The optimal phase difference clearly depends on the frequency, moving from $N \cdot \Delta\phi = 0.5$ ("C" shape undulation) for $\nu = 0.2$ Hz to $N \cdot \Delta\phi = 1.0$ ("S" shape undulation) for $\nu = 1.0$ Hz. Snapshots from the video of the locomotion producing the maximal crawling speed are visible in figure 9.

The complete results are presented in figure 7. Running the robot with $A = \pm 10°$ produces practically no



(a)$\nu = 0.2$ Hz

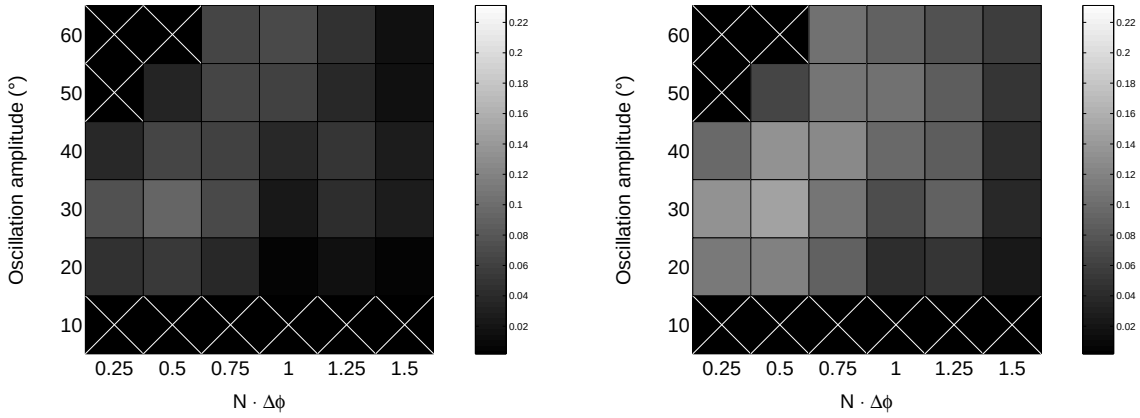
(b)$\nu = 0.4$ Hz

(c)$\nu = 0.6$ Hz

(d)$\nu = 0.8$ Hz

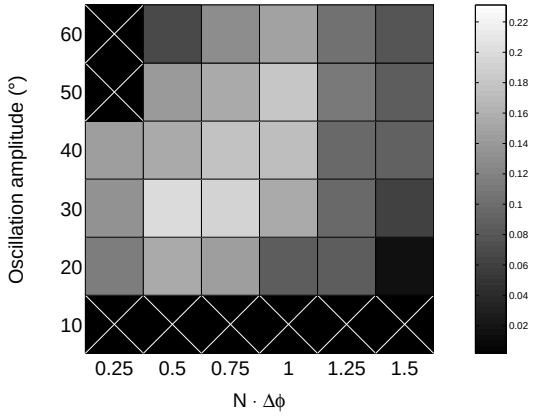
(e)$\nu = 1.0$ Hz

Fig. 7. Crawling locomotion speeds (in m/s) with the different parameters

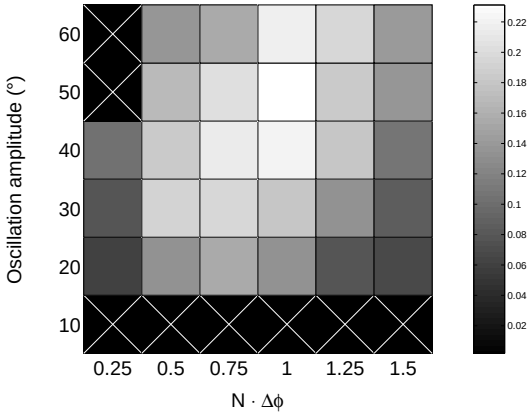


(a)$\nu = 0.2$ Hz

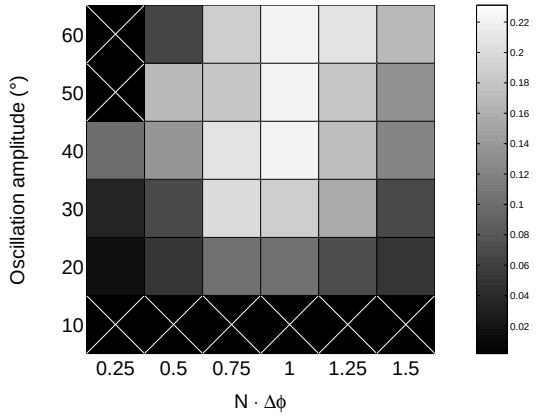
(b)$\nu = 0.4$ Hz



(c)$\nu = 0.6$ Hz



(d)$\nu = 0.8$ Hz



(e)$\nu = 1.0$ Hz

Fig. 8. Swimming locomotion speeds (in m/s) with the different parameters

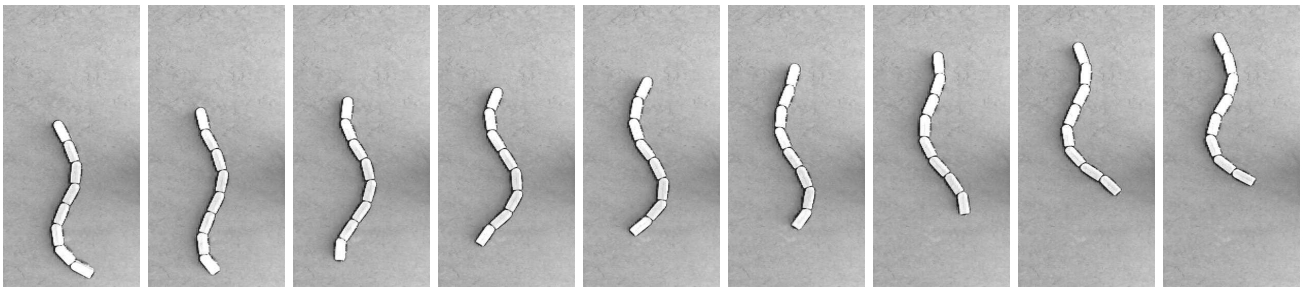


Fig. 9. The robot crawling at $A = \pm 30°$, $N \cdot \Delta\phi = 1.0$ and $\nu = 1.0$ Hz. The time step between the snapshots is 0.12 s.

locomotion, as the passive wheels tend to bend around their axis and slip on the ground. The optimum is clearly peaked, and there are no local maxima. This is interesting, as it suggests that relatively simple online optimization algorithms could be used to adapt the CPG parameters to the current environment in which the robot is moving.

Compared to the simulation, the obtained maximal speed is 1.2 times larger. The difference is mostly due to the fact that the simulated friction model is only an approximation of the passive wheels used on the robot. Furthermore, the speed of locomotion is sensitive to friction coefficient values. For example, a change of the μ_2 friction coefficient of the simulation from 0.025 to 0.05 decreases the maximum obtained speed to 0.25 m/s (i.e. a drop of 24%). Another difference is that with the real robot, the optimal phase was correlated to the frequency; this effect was not visible in simulation and is likely to have the same origin of the observed speed difference (i.e. the approximated friction model).

C. Swimming with the real robot

The speed for a given set of parameters was measured as follows: the robot was started at the beginning of the aquarium. A chronometer was started when it reached a first horizontal line placed at 1.15 m from the border, and stopped when crossing the second line, placed at 1.00 m from the first one. For some parameters producing very low speeds, the chronometer has been stopped before the second line and the distance manually measured. No measure has been taken for $A = \pm 10°$, as the robot did not stay vertically in this configuration, due to small asymmetries of the center of mass.

Depending on the parameters used, the swimming speed varied between 0 and 0.23 m/s (0.30 body lengths/s). The optimal amplitude was $A = \pm 30°$ at low frequencies, moving to $A = \pm 50°$ for $\nu \geq 0.8$ Hz. Similarly, the optimal phase moves with the frequency: the optima at $\nu < 0.8$ Hz have $N \cdot \Delta\phi = 0.5$, and those at $\nu \geq 0.8$ Hz have $N \cdot \Delta\phi = 1.0$. The complete results are plotted in figure 8. The optimum is peaked, although less remarkably than in crawling. Snapshots from the video of the locomotion producing the maximal swimming speed are visible in figure 10.

Compared to the results of the simulation, the maximal swimming speed appears to be almost 4 times lower; this is not really surprising, as a really simplified hydrodynamic force model has been used and more work is needed to calibrate the drag coefficient values. Even if the position of the optimum is not the same, especially at high frequencies, the structure of the results is similar, although the existence of small local maxima at $\nu < 0.6$ Hz has not been predicted by the simulation and is maybe the effect of turbulence induced by the robot when swimming.

VI. Future work

- The simulation has to be more carefully calibrated to the real robot, in order to obtain quantitatively valid results (especially for swimming). The hydrodynamics force model also needs to be improved.
- We plan to implement online optimization of the CPG parameters (e.g. using Powell's method), therefore allowing the robot to adapt to a new environment without needing to know in advance the optimal CPG parameters (in terms of obtained speed). The structure of the results obtained with characterization and some preliminary simulations seem to show that this is possible.

VII. Conclusions

In this paper, we presented AmphiBot II, an amphibious snake robot with an on-board CPG-based trajectory generator. The main contribution of this article is a detailed characterization of how the CPG parameters (i.e., amplitude, frequency and wavelength) influence the locomotion speed of the robot. The structure of the observed results predicts the possible usage of online optimization methods to find a set of parameters maximizing the speed in any environment.

Acknowledgments

We acknowledge the support from Francesco Mondada and the Autonomous Systems Laboratory (ASL) at EPFL, for their PD motor controller. This research is funded by a Young Professorship Award to Auke Ijspeert from the Swiss National Science Foundation.

References

[1] Y. Umetani and S. Hirose, "Biomechanical study of active cord mechanism with tactile sensors," in *Proceedings of the 6th international symposium on industrial robots*, Nottingham, 1976, pp. c1-1–c1-10.

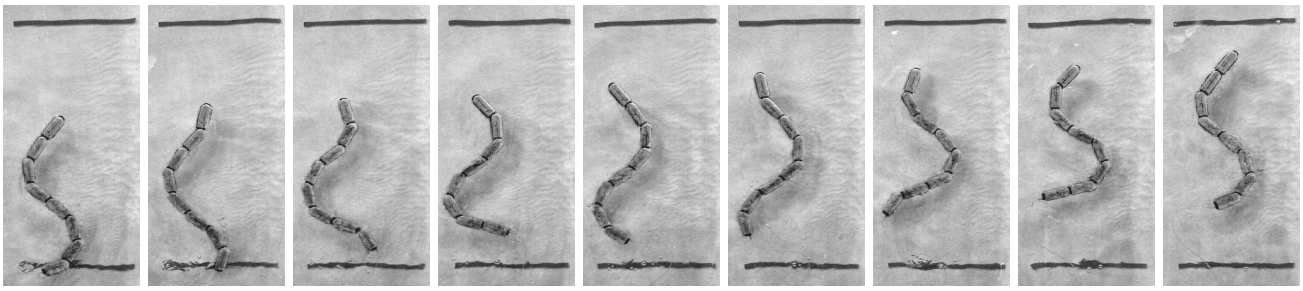


Fig. 10. The robot swimming at $A = \pm 50°$, $N \cdot \Delta\phi = 1.0$ and $\nu = 0.8$ Hz. The time step between the snapshots is 0.16 s.

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

[3] T. Lee, T. Ohm, and 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, Texas, 1994, pp. 142–149.

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

[5] B. Klaassen and 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*. 1999, pp. 3014–3019, IEEE.

[6] G.S.P. Miller, *Neurotechnology for biomimetic robots*, chapter Snake robots for search and rescue, Bradford/MIT Press, Cambridge London, 2002.

[7] H.R. Choi and S.M. Ryew, "Robotic system with active steering capability for internal inspection of urban gas pipelines," *Mechatronics*, vol. 12, pp. 713–736, 2002.

[8] K. Dowling, *Limbless Locomotion: Learning to Crawl with a Snake Robot*, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1997.

[9] R. Worst, "Robotic snakes," in *Third German Workshop on Artifical Life*. 1998, pp. 113–126, Verlag Harri Deutsch.

[10] K.A. McIsaac and 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*. 1999, pp. 2843–2848, IEEE.

[11] C. Wilbur, W. Vorus, Y. Cao, and S.N. Currie, *Neurotechnology for biomimetic robots*, chapter A Lamprey-Based Undulatory Vehicle, Bradford/MIT Press, Cambridge London, 2002.

[12] T. Takayama and S. Hirose, "Amphibious 3D active cord mechanism "HELIX" with helical swimming motion," in *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2002, pp. 775–780, IEEE.

[13] A. Crespi, A. Badertscher, A. Guignard, and A.J. Ijspeert, "AmphiBot I: An amphibious snake-like robot," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 163–175, 2005.

[14] A. Crespi, A. Badertscher, A. Guignard, and Ijspeert. A.J., "Swimming and crawling with an amphibious snake robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, 2005, pp. 3035–3039.

[15] F. Delcomyn, "Neural basis for rhythmic behaviour in animals," *Science*, vol. 210, pp. 492–498, 1980.

[16] A.J. Ijspeert, A. Crespi, and J.M. Cabelguen, "Simulation and robotics studies of salamander locomotion. Applying neurobiological principles to the control of locomotion in robots," *Neuroinformatics*, vol. 3, no. 3, pp. 171–196, 2005.

[17] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[18] J. Conradt and P. Varshavskaya, "Distributed central pattern generator control for a serpentine robot," in *ICANN 2003*, 2003.