Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model

Auke Jan Ijspeert and Alessandro Crespi

***Abstract*— This article presents a control architecture for controlling the locomotion of an amphibious snake/lamprey robot capable of swimming and serpentine locomotion. The control architecture is based on a central pattern generator (CPG) model inspired from the neural circuits controlling locomotion in the lamprey's spinal cord. The CPG model is implemented as a system of coupled nonlinear oscillators on board of the robot. The CPG generates coordinated travelling waves in real time while being interactively modulated by a human-operator. Interesting aspects of the CPG model include (1) that it exhibits limit cycle behavior (i.e. it produces stable rhythmic patterns that are robust against perturbations), (2) that the limit cycle behavior has a closed-form solution which provides explicit control over relevant characteristics such as frequency, amplitude and wavelength of the travelling waves, and (3) that the control parameters of the CPG can be continuously and interactively modulated by a human operator to offer high maneuverability. We demonstrate how the CPG allows one to easily adjust the speed and direction of locomotion both in water and on ground while ensuring that continuous and smooth setpoints are sent to the robot's actuated joints.**

I. INTRODUCTION

Online trajectory generation for robots with multiple degrees of freedom is still a difficult and unsolved problem. The control of locomotion, for instance, requires multi-dimensional coordinated rhythmic patterns that need to be correctly tuned such as to satisfy multiple constraints: the capacity to generate forward motion, with low energy, without falling over, while adapting to possibly complex terrain (uneven ground, obstacles), and while allowing the modulation of speed and direction. In vertebrate animals, an essential building block of the locomotion controller is the *central pattern generator* (CPG) located in the spinal cord. A CPG is a neural circuit capable of producing coordinated patterns of rhythmic activity in open loop, i.e. without any rhythmic inputs from sensory feedback or from higher control centers [1], [2]. Interestingly, very simple input signals are sufficient to modulate the produced patterns. In a decerebrated cat for instance, increasing the strength of a simple electrical stimulation to the CPG will lead to an increase of the frequency of oscillations as well as switches between different gaits from walk to trot to gallop [3]. In the lamprey, speed of swimming can similarly be adjusted by the level of an electrical stimulation, while stimulating the spinal cord with different strengths between left and right leads to turning behavior [4]. From a control point of view, CPGs therefore implement some kind of feedforward controller, i.e. a controller that "knows" which torques need to be rhythmically applied to obtain a given speed of locomotion. Interestingly, CPGs combine notions of stereotypy (steady state locomotion tends to show little variability) and of flexibility (speed, direction and types of gait can continuously be adjusted).

In this article, we use a CPG model inspired from the lamprey to control the swimming and serpentine locomotion of an amphibious snake robot. Our goal is to demonstrate that the CPG implemented as a system of coupled nonlinear oscillators is an ideal building block for doing online trajectory generation in a redundant robotic system. We therefore designed a control architecture in which a CPG model is programmed onboard of our amphibious robot and continuously receives high-level commands wirelessly from a human operator to modulate the speed and direction of locomotion both in water and on ground. We thus obtain a system that is interactive (i.e. with a human-in-the-loop), generates trajectories in real-time, and offers high maneuverability in water and on ground.

A. Related work

A variety of snake-like robots have been constructed. Most of them were designed for use on ground [5], [6], [7], [8], [9], a few were designed for swimming [10], [11], and even fewer for both water and ground [12], [13]. Their control architecture can roughly be divided into three categories: sine-based, model-based, and CPG-based.

Sine-based approaches use simple sine-based functions for generating travelling waves (see for instance [7], [9]). The advantages of such an approach are its simplicity and the fact that important quantities such as frequency, amplitude and wavelength are explicitly defined. A disadvantage is that online modifications of the parameters of the sine function (e.g. the amplitude or the frequency) will lead to discontinuous jumps of setpoints, which will generate jerky movements with risks of damaging the motors and gearboxes. This problem can to some extent be overcome by filtering the parameters and/or the outputs but the approach then loses its simplicity. Another disadvantage is that sine-based functions do not offer simple ways of integrating sensory feedback signals.

Model-based approaches use kinematic [14], [15] or dynamic [16], [17], [18], [19] models of the robot to design control laws for gait generation. The control laws are sometimes based on sine-based functions as above (e.g. [14], [19]), but the model-based approaches offer a way to identify

A. Crespi and A.J. Ijspeert are with the School of Computer and Communication Sciences, EPFL (Ecole Polytechnique Fédérale de Lausanne), CH 1015 Lausanne, Switzerland alessandro.crespi@epfl.ch, auke.ijspeert@epfl.ch

fastest gaits for a given robot by using kinematic constraints or approximations of the equations of motion, for instance. Model-based approaches are therefore very useful for helping to design controllers but have two limitations. First, the resulting controllers are not always suited for interactive modulation by a human operator. Second, the performance of controllers will deteriorate when models become inaccurate, which is rapidly the case for interaction forces with a complex environment (e.g. friction with uneven ground).

CPG-based approaches use dynamical systems, e.g. systems of coupled nonlinear oscillators or recurrent neural networks, for generating the travelling waves necessary for locomotion (see for instance [20], [21], [11], [22]). These approaches are implemented as differential equations integrated over time, and the goal is to produce the travelling wave as a limit cycle. If this is the case, the oscillatory patterns are robust against transient perturbations (i.e. they asymptotically return to the limit cycle). Furthermore, the limit cycle can usually be modulated by some parameters which offer the possibility to smoothly modulate the type of gaits produced. Finally, CPGs can readily integrate sensory feedback signals in the differential equations, and show interesting properties such as entrainment by the mechanical body [23].

However, one difficulty with CPG-based approaches is how to design the CPG to produce a particular pattern. Many CPG models do not have explicit parameters defining quantities such as frequency, amplitude, and wavelength (for instance, a van der Pol oscillator does not have explicit frequency and amplitude parameters). This does not need to be the case. In this article, we use a CPG model based on amplitude-controlled phase oscillators. An interesting aspect of this approach is that the limit cycle of the CPG has a closed form solution, with explicit frequency, amplitude and wavelength parameters. The approach therefore combines the elegance and robustness of the CPG approaches with the simplicity of sine-based approaches. Furthermore, our CPG model is computationally very light which makes it well suited to be programmed on a simple microcontroller on board of the robot. The implementation of the CPG is inspired from lamprey models [24]. It is close to the CPG model presented in [22], but differs in the following aspects: (1) it is made of a double chain of oscillators, (2) it has differential equations controlling the amplitudes of each oscillator (not only the phase), (3) it has an interface function that allows easy modulation of speed and direction by a human operator, and (4) the CPG is used to control not only serpentine crawling but also swimming.

B. Outline of the article

In the rest of the article, we will first briefly describe the mechanical and electronic hardware of Amphibot II, our amphibious snake robot (Section II). We then present the CPG model and the control architecture (Section III). Based on systematic exploration of the travelling waves that lead to the fastest locomotion on ground and in water, we define an interface for modulating the speed and direction of locomotion. In Section IV, we present results that characterize how the CPG and the interface allows us to control the locomotion of the robot. Examples of interactive locomotion, i.e. locomotion that is continuously modulated by a human operator, are also shown. We conclude the article with a short discussion and presentation of future work (Section V).

Fig. 1. The AmphiBot II robot.

II. AMPHIBOT II: AN AMPHIBIOUS SNAKE/LAMPREY ROBOT

The AmphiBot II robot has a modular design and is constructed out of 7 actuated elements and a head element (which is externally identical to the other elements but has no motor). The external casing of each element consists of two symmetrical parts molded using polyurethane resin. The elements are connected (both mechanically and electrically) using a compliant connection piece fixed to the output axis, which contains 6 wires. All the output axes of the elements are aligned, therefore producing planar locomotion. To ensure the waterproofing of the robot, custom O-rings are used.[1] The total length of the robot is 77 cm. 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 strong Velcro. Currently the wheels are removed for swimming, except for experiments in which we do transitions between water and ground. 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.

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 512-pulse 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).

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. The motor controller is based on a PIC16F876A microcontroller, which runs a PID motor controller developed at the Autonomous Systems Laboratory of the EPFL. The battery has a capacity of 600 mAh, and can

[1]During extensive swimming tests, air is insufflated inside the robot by a small pump through a highly flexible PVC tube for maintaining a little overpressure inside the elements and avoiding leakage.

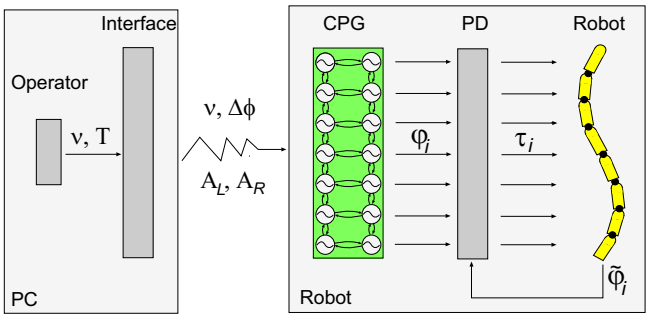

Fig. 2. Different components of the locomotion controller. High level speed and turning commands ν and T, respectively, are provided by a human operator via a keyboard or a joystick connected to a PC. An interface on the PC transforms those commands into the control parameters of the CPG ($\nu, \Delta\phi, A_L$ and A_R) and sends them wirelessly to the robot. The CPG model on board of the robot generates desired trajectories φ_i for each motor and passes them to standard PD controllers which produce the torques τ_i. $\tilde{\varphi}_i$ are the actual angles generated by the actuated joints.

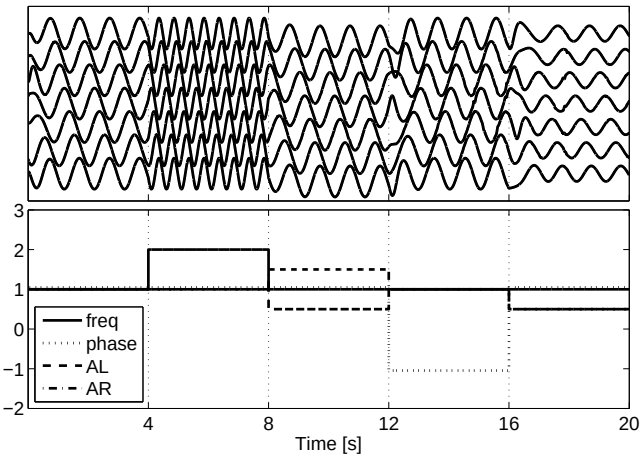

Fig. 3. Effect of changing the parameters of the CPG. Top: setpoint signals, Bottom: control parameters. Initial parameters are A_L=A_R=1, ν=1 Hz and $N \cdot \Delta\phi$=1. At t=4s, ν is temporarily changed to 2.0 Hz, at t=8s, A_L and A_R are temporarily changed to 0.5 and 1.5 respectively which leads to a negative offset of the setpoint oscillations. At t=12s, $N \cdot \Delta\phi$ is temporarily set to -1.0 which leads to a reversal of the direction of the travelling wave. At t=16s, A_L and A_R are changed to 0.5 which leads to reduced amplitude in the oscillations.

power an element for approximately two hours of continuous use in normal conditions. A water detector circuit is used internally to detect any leakage (blinking LED).

B. Head element

The head element has a PIC18F2580 microcontroller, which is master on the I^{2}C bus of the robot and which runs the CPG model. It sends out the setpoints to the motor controllers of each element in realtime. The microcontroller also communicates with a PIC16LF876A microcontroller, which controls a nRF905 radio transceiver. 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 water up to at least 30 cm (the maximum tested depth). The more common 2.4 GHz band has not been used 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 from the PC to the online trajectory generator.

III. LOCOMOTION CONTROL

The general architecture of the locomotion control scheme is shown in Fig. 2. The control scheme is implemented partly on a PC which takes the speed and direction commands from a human operator, transforms them into CPG control parameters, and send them wirelessly to the robot, and partly on the robot which has a CPG model on board for online trajectory generation and PD controllers for transforming the desired trajectories into torques applied to the actuated joints.

A. Central pattern generator model

Our CPG model is based on a system of amplitude-controlled phase oscillators. The structure of the CPG is inspired from the lamprey and forms a double chain of oscillators with nearest neighbor coupling (Fig. 2). 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. Like in the real lamprey, turning can be obtained by inducing higher amplitude oscillations on one side of the double chain (see below). Compared to previous neural network models that we developed of the lamprey CPG [25], [26], the model in this article is simpler (much fewer state variables) and therefore better suited for being programmed on a microcontroller on board of the robot, while keeping the essential features of lamprey travelling wave generation.

The total number of oscillators is $2N$, where $N = 7$ is the number of actuated joints in the robot. Actuated joints are numbered 1 to N from head to tail. Oscillators in the left chain of the CPG are numbered 1 to N and those on the right side are numbered N+1 to $2N$ from head to tail.

The CPG is implemented as the following system of $2N$ coupled oscillators:

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

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}. The variable x_i is the rhythmic and positive output signal extracted out of oscillator i. The first differential equation determines the time evolution of the phase θ_i. It can be shown that two (or more) coupled oscillators will synchronize (i.e. oscillate at the same frequency and with a constant phase lag) if the coupling weights w_{ij} are sufficiently large compared to the differences of intrinsic frequencies (see Appendix). The phase lag between the oscillators will then depend on ϕ_{ij}, w_{ij}

and ν_i. The second differential equation is a second order linear differential equation that ensures that the amplitude r_i smoothly converges to R_i in a critically dampened fashion.

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

$$\begin{aligned} \varphi_i &= x_i - x_{N+1} \\ \tau_i &= K_p e_i + K_d \dot{e}_i \end{aligned} \quad (2)$$

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.

To reflect the symmetries of the robot, we set several parameters to the same values. The frequency parameters are equal for all oscillators, i.e. $\nu_i = \nu$. We also chose all amplitude parameters on one side of the CPG to be equal: $R_i = A_L$ for the left side ($i = [1, ..., N]$) and $R_i = A_R$ for the right side ($i = [N+1, ..., 2N]$). The phase biases ϕ_{ij} are 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. The parameter $\Delta\phi$ will determine the phase lag between modules, see below. We used $w_{ij} = 4$ for all connections and $a_i = 100$ for all oscillators. 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).

With these settings, the CPG asymptotically converges to a limit cycle that is defined by the following closed form solution for the i^{th} actuated joint (a skeleton of the proof is given in Appendix):

$$\varphi_i^{\infty}(t) = A_L - A_R + (A_L + A_R) \cdot \cos(2\pi\nu \cdot t + i\Delta\phi + \phi_0) \quad (3)$$

where ϕ_0 depends on the initial conditions of the system. This means that the system always stabilizes into a travelling wave which depends on the four control parameters $\nu, \Delta\phi, A_L$ and A_R. Indeed the frequency, phase lag, amplitude and offset are directly determined by ν, $\Delta\phi$, A_L+A_R, and A_L-A_R, respectively. These parameters 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 wave after a short transient period. An example of how the CPG reacts to parameter changes can be observed in Fig. 3: when the parameters are changed, the oscillator smoothly converges to the new limit cycle, without any discontinuities in the outputs.

The differential equations are 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.

B. Interface for the control parameters

To simplify the control of the robot by a human operator, it is useful to reduce the number of commands to two, one for speed and one for direction, instead of the four control parameters for the CPG.[2]

Turning can be induced by modulating A_L-A_R, i.e. by adding offsets to the setpoint oscillations. The robot will then make undulations around a bent posture and turn towards the side with higher amplitude. We can therefore introduce the turning command T which determines the difference between left and right amplitudes normalized by the total amplitude, namely $T = \frac{A_L - A_R}{A_L + A_R}$.

The control of speed is more difficult because the speed of locomotion depends jointly on the frequency ν, the amplitude A=A_L+A_R and the phase lag $\Delta\phi$ of the travelling wave, as well as on the type of environment (e.g. the type of friction with the ground, the slope, etc. ...). In [27], we carried out a systematic exploration of how the speed of locomotion depends on these three parameters for two different environments, a flat wooden floor and a small pool with water. The parameters have been kept into a reasonable range: the amplitude A=A_L+A_R between 10° and 60° (with a step of 10°), the frequency ν between 0.2 Hz and 1.0 Hz (with a step of 0.2 Hz) and the phase difference $\Delta\phi$ between $0.25/N$ and $1.5/N$ (with a step of $0.25/N$).

The outcome of that study is that, in the explored parameter space, the speed of locomotion always monotonically increases with the frequency when the two other parameters are kept fixed at any value. The amplitude and phase lag show a more complex, non-monotonic, influence on the speed. For a given frequency, for instance, the dependence of speed on the amplitude and phase parameters is a smooth function with a single optimum. The location of the optimum varies with the frequency. For instance, on ground with $\nu = 0.2Hz$ the maximum speed (0.15 m/s) is obtained with A=30° and $\Delta\phi$=0.5/N, while at $\nu = 1.0Hz$ the maximum speed (0.40 m/s) is obtained with A=30° and $\Delta\phi$=1.0/N. In order words, with our robot it is better to make C-shaped undulations ($\Delta\phi$=0.5/N) at low frequencies and S-shaped undulations ($\Delta\phi$=1.0/N) at higher frequencies. The same is true for swimming. See [27] for all the data and more detailed analysis.

We therefore choose the frequency as a single command parameter for speed, and design two functions $[A, \Delta\phi] = f_{ground}(\nu)$ and $[A, \Delta\phi] = f_{water}(\nu)$ for setting the amplitude and phase lag for a given frequency. These piecewise linear functions are simple linear interpolations between the observed optima. The functions thus ensure that the travelling wave produced at a given frequency remains close to the fastest locomotion at that frequency. When the robot makes a transition from ground to water or vice-versa, the human operator makes a manual switch from one function to the other.[3] Note that, since these functions depend on the environment, we will in the future use online optimization to identify good parameters for a given (possibly unknown) environment instead of systematic search (see Section V).

[2]Note that by design the robot is only capable of planar locomotion and therefore does not require control of vertical motion.

[3]This switch will soon be done automatically using an external water sensor.

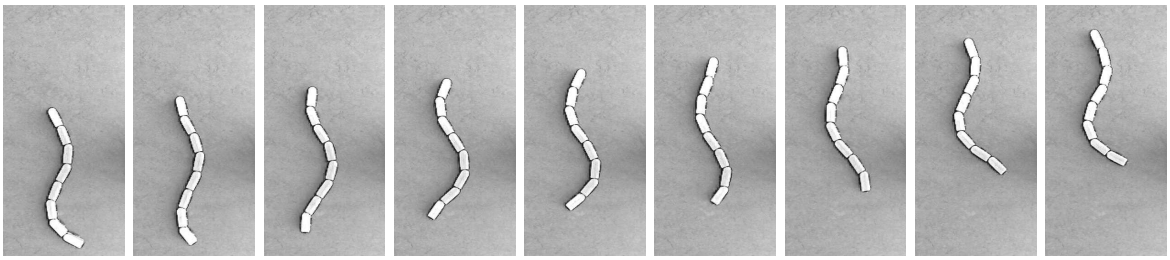

Fig. 4. The robot crawling at $A = \pm 30°$, $N \cdot \Delta\phi = 1.0$ and $\nu = 1.0$ Hz. The time step between the snapshots is 0.12 s. See also the accompanying video.

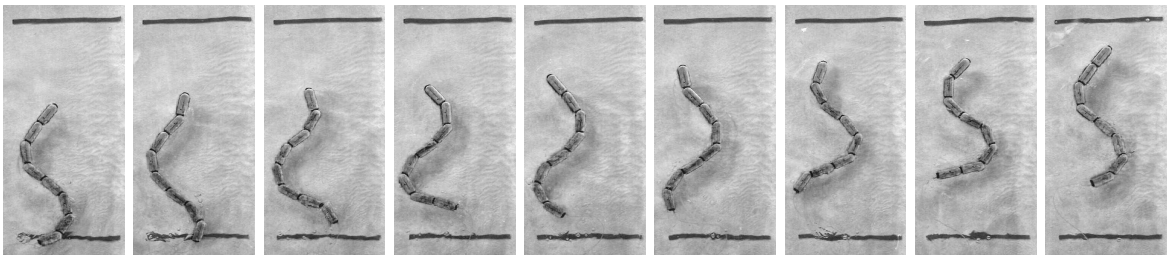

Fig. 5. The robot swimming at $A = \pm 50°$, $N \cdot \Delta\phi = 1.0$ and $\nu = 0.8$ Hz. The time step between the snapshots is 0.16 s. See also the accompanying video.

With this interface, the speed and direction of locomotion of the snake robot can now easily be adjusted in real-time by a human operator by setting the frequency ν and the turning command T.

IV. EXPERIMENTAL RESULTS

A. Typical swimming and serpentine gaits

Figures 4 and 5 show snapshots of the robot doing serpentine locomotion on ground and anguilliform swimming in water. In both cases, the undulation of body deformations travelling from head to tail propels the robot forward because of the asymmetrical interaction forces with the environment, namely low drag/friction in the longitudinal direction, and high drag/friction in the perpendicular direction.

B. Control of speed during serpentine locomotion

We systematically tested the speed of serpentine locomotion using different values of our command parameter ν. Figure 6 shows the resulting values. The speed is evaluated by measuring the time needed by the robot to travel a given distance (1m), and repeating the measure four times.

Results show that the speed increases monotonically with ν. The highest speed at 1.0Hz is approximately 0.4m/s, which corresponds to 0.55 bodylength/s. Higher speeds can be reached at higher frequencies, but tracking errors in the PD controllers become significative above 1.0Hz due to motor torque limits. As explained in Section III-B, because of the function f_{ground}, the types of undulations are quite different between low frequencies where the undulations make C-shapes ($\Delta\phi$=0.5/N) and high frequencies where the undulations make S-shapes ($\Delta\phi$=1.0/N). Note that while the speed measures have been made at fixed ν values, ν can be continuously and interactively adjusted to produce locomotion with smooth accelerations and decelerations.

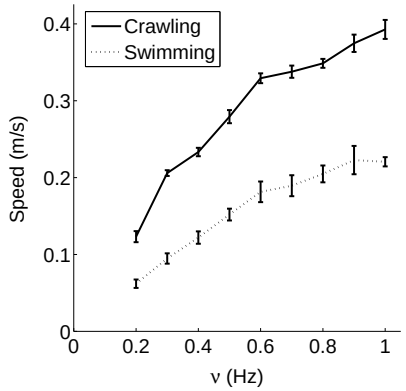

Fig. 6. Control of speed during serpentine crawling (continuous line) and swimming (dotted line). Each data point is the average of 4 speed measures, and the error bars correspond to the standard deviation.

C. Control of direction during serpentine locomotion

To evaluate the turning ability of the robot on ground, we used video tracking of a green LED mounted on the head of the robot. When a non zero turning command T is sent to the robot, it will on average progress on a circle. Figure 7 shows the trace that the head element makes. A circle is fitted to the outer bounds of the trace to provide an estimation of the turning ability: the shorter the radius R, the sharper the turning. Figure 8 shows how the inverse of the radius varies with the T command. Interestingly, the relation between $1/R$ and T is almost linear. The sharpest turning is obtained at T=1, where the radius of the curvature is 25cm. Turning is therefore quite sharp for a 72cm long robot.

D. Control of speed during swimming

Similarly to locomotion on ground, we tested how the speed of swimming depends on the command ν. Speed was measured by taking the time necessary to travel a given

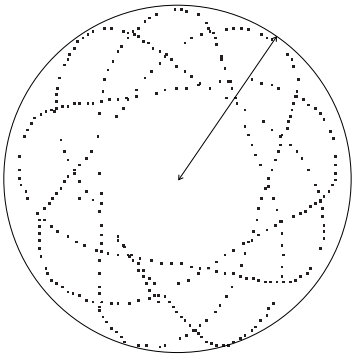

Fig. 7. Tracking of the robot while turning on ground. The dotted line is the trace left by the head element. The radius of the circle fitting the outer bounds is used to measure the curvature.

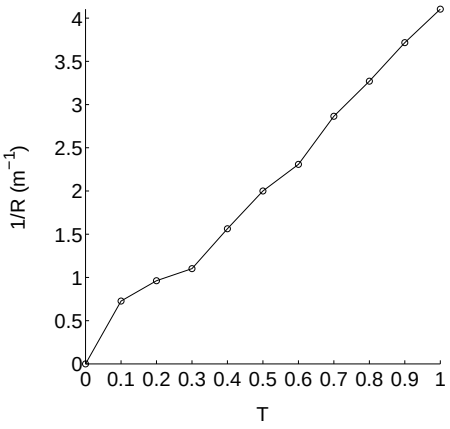

Fig. 8. Control of direction during serpentine locomotion. The horizontal and vertical axes are respectively the turning command T and the inverse of the radius 1/R.

distance. Since accelerations are slower in water than on ground, we waited enough time (approx 5 seconds) before the beginning of the measurement such as to be close to steady-state swimming. Figure 6 shows the results of the measurements. Speed increases monotonically with ν up to ν=0.9Hz where it saturates. Maximum speed of 0.23m/s, i.e. 0.32 bodylengths/s, are attained. Compared to serpentine locomotion, the speeds are lower and the measurements show a larger variability. The larger variability is related to the fact that water in motion makes experiments less reproducible because of the complex dynamics of waves bouncing against the small swimming pool windows.

E. Control of direction during swimming

Turning in water is induced like on ground by modulating the turning command T. The robot can turn on a circle that is less than 40cm of diameter (our testing pool is 80cm wide). See the accompanying video. We have not yet tested how the curvature varies with different T values because we do not yet have access to a sufficiently large swimming pool with overhead camera. Such tests will be done in the near future.

F. Remotely operated robot

One of the main motivations behind our CPG-based control architecture is to allow high maneuverability and interactive locomotion control with a human-in-the-loop. We therefore tested the robot on ground, in water, and with transitions between the two by continuously remote controlling the robot via the commands ν and T. As shown in the accompanying video, the robot is capable of continuously accelerating, decelerating, and changing directions. Because of the simple speed and direction commands, the operation of the robot is intuitive and accessible to any operator without prior training.

V. CONCLUSIONS AND FUTURE WORK

We presented an amphibious snake robot capable of swimming and crawling controlled by a central pattern generator (CPG) model. The CPG model is designed to produce travelling waves as limit cycle behavior, and to allow simple modulation of the frequency, amplitude, and phase lag of the travelling undulations. Based on systematic exploration of the speed of locomotion on ground and in water, we designed interface functions to allow a human operator to continuously adjust the speed and direction of locomotion while ensuring that the produced travelling waves lead to the fastest locomotion for a given frequency in the two tested environments. This work therefore demonstrates that a CPG model is a useful building block for solving the problem of trajectory generation in redundant system and for allowing high maneuverability.

We are currently extending this work by doing online optimization of the interface functions. The bad news of this work and the study presented in [27], is that, for a given frequency, the speed of locomotion varies significantly with the chosen amplitude and phase lag of the undulation. In order words, it is important to identify the optimal parameters leading to fastest locomotion, since moving away from them leads to significantly slower locomotion. Furthermore, the optimal parameters vary from one frequency to the other, and from one medium to the other (e.g. it changes when there is a slope or when the friction properties of the ground change). The good news is that the function relating speed to the amplitude and phase lag is smooth and has a single global optimum (in the parameter space that we studied). It would therefore be quite simple to find the optimum of that function using standard optimization algorithms (e.g. Simplex or Powell's method). We are therefore exploring how the robot could continuously track the optimal undulation for a given environment and frequency. This can be done in real-time without human supervision and without needing to stop the operation of the robot, since the CPG will keep running while the parameter space is explored.

Acknowledgments This work was supported by the Swiss National Science Foundation. We acknowledge the support from Francesco Mondada and the Autonomous Systems Laboratory (ASL) at EPFL, for their PD motor controller.

VI. APPENDIX

The limit cycle of the CPG is determined by the time evolution of the amplitude and phase variables. We here show the particular case of two oscillators coupled bi-directionally:

$$\begin{array}{rcl} \dot{\theta}_1 & = & 2\pi\nu_1 + w\sin(\theta_2 - \theta_1 - \Delta\phi) \\ \ddot{r}_1 & = & a(\frac{a}{4}(R_1 - r_1) - \dot{r}_1) \\ \dot{\theta}_2 & = & 2\pi\nu_2 + w\sin(\theta_1 - \theta_2 + \Delta\phi) \\ \ddot{r}_2 & = & a(\frac{a}{4}(R_2 - r_2) - \dot{r}_2) \end{array} \quad (4)$$

It is easy to demonstrate that the state variables r_1 and r_2 asymptotically converge to R_1 and R_2, respectively, from any initial condition. Since we are interested in determining whether these two oscillators will synchronize (i.e., evolve with a constant phase difference), and, if yes, with which phase difference, it is useful to introduce the phase difference ψ=θ2-θ1. The time evolution of the phase difference is determined by

$$\dot{\psi} = f(\psi) = \dot{\theta}_2 - \dot{\theta}_1 = 2\pi(\nu_2 - \nu_1) - 2w\sin(\psi - \Delta\phi) \quad (5)$$

If the oscillators synchronize, they will do so at the fixed points ψ_∞ (i.e., points where $f(\psi_\infty)$ = 0)):

$$\psi_\infty = \arcsin(\frac{\pi(\nu_2 - \nu_1)}{w}) + \Delta\phi \quad (6)$$

In our case we have ν_1=ν_2=ν, and this equation has a single solution ψ_∞=$\Delta\phi$. This solution is asymptotically stable because $\partial f(\psi_\infty)/\partial\psi < 0$. The outputs of the oscillators therefore asymptotically converge to oscillations that are phase-locked with a phase difference of $\Delta\phi$: $x_1^\infty(t) = R_1(1 + \cos(2\pi\nu t + \phi_0))$ and $x_2^\infty(t) = R_2 1 + \cos(2\pi\nu t + \Delta\phi + \phi_0))$ where ϕ_0 is a constant that depends on initial conditions. Since the complete CPG is made of multiple bi-directionally coupled oscillators and that all parameters ϕ_{ij} are consistent (i.e. the sums of the parameters ϕ_{ij} are equal to a multiple of 2π on any closed path between oscillators), the same reasoning can be recursively applied to demonstrate convergence of the complete CPG.

REFERENCES

[1] F. Delcomyn. Neural basis for rhythmic behaviour in animals. *Science*, 210:492–498, 1980.

[2] S. Grillner. Neural control of vertebrate locomotion – central mechanisms and reflex interaction with special reference to the cat. In W.J.P. Barnes and Gladden M.H., editors, *Feedback and motor control in invertebrates and vertebrates*, pages 35–56. Croom Helm, 1985.

[3] M.L. Shik, F.V. Severin, and G.N. Orlovsky. Control of walking by means of electrical stimulation of the mid-brain. *Biophysics*, 11:756–765, 1966.

[4] M.G. Sirota, G. Viana Di Prisco, and R. Dubuc. Stimulation of the mesencephalic locomotor region elicits controlled swimming in semi-intact lampreys. *European Journal of Neuroscience*, 12:4081–4092, 2000.

[5] G.S. Chirikjian and J.W. Burdick. Design, implementation, and experiments with a thirty-degree-of-freedom 'hyper-redundant' robot. In *4th International Symposium on Robotics and Manufacturing (ISRAM 1992)*, 1992.

[6] B. Klaassen and K.L. Paap. GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1999)*, pages 3014–3019. IEEE, 1999.

[7] G.S.P. Miller. Snake robots for search and rescue. In J. Ayers, J.L. Davis, and A. Rudolph, editors, *Neurotechnology for biomimetic robots*. Bradford/MIT Press, Cambridge London, 2002.

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

[9] D.P. Tsakiris, M. Sfakiotakis, A. Menciassi, G. La Spina, and Dario P. Polychaete-like undulatory robotic locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 3029–3034. 2005.

[10] K.A. McIsaac and J.P. Ostrowski. A geometric approach to anguilliform locomotion: Simulation and experiments with an underwater eel-robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1999)*, pages 2843–2848. IEEE, 1999.

[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] H. Yamada, S. Chigisaki, M. Mori, K. Takita, K. Ogami, and Hirose S. Development of Amphibious Snake-like Robot ACM-R5. In *The Proceedings of the 36th International Symposium on Robotics*, 2005.

[13] A. Crespi, A. Badertscher, A. Guignard, and A.J. Ijspeert. *AmphiBot I*: an amphibious snake-like robot. *Robotics and Autonomous Systems*, 50–4:163–175, 2005.

[14] J. Ostrowski and J. Burdick. Gait kinematics for a serpentine robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1996)*, pages 1294–1299. IEEE, April 1996.

[15] F. Matsuno and K. Suenaga. Control of redundant 3D snake robot based on kinematic model. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA 2003)*, pages 2061–2066. IEEE, September 2003.

[16] P. Prautsch and T. Mita. Control and analysis of the gait of snake robots. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, pages 502–507. IEEE, August 1999.

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

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

[19] K. McIsaac and J. Ostrowski. Motion planning for anguilliform locomotion. *IEEE Transactions on Robotics and Automation*, 19(4):637–652, 2003.

[20] Z. Lu, B. Ma, S. Li, and Y. Wang. Serpentine Locomotion of a Snake-like Robot Controlled by Cyclic Inhibitory CPG Model. In *The Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pages 96–101, 2005.

[21] D.P. Tsakiris, M. Sfakiotakis, and A. Vlakidis. Biomimetic centering for undulatory robots. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2006)*, pages 744–749. 2006.

[22] J. Conradt and P. Varshavskaya. Distributed central pattern generator control for a serpentine robot. In *International Conference on Artificial Neural Networks (ICANN 2003)*, 2003.

[23] G Taga. A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics*, 78(1):9–17, 1998.

[24] A.H. Cohen, P.J. Holmes, and R. Rand. The nature of coupling between segmented oscillations and the lamprey spinal generator for locomotion: a mathematical model. *J. Math. Biol.*, 13:345–369, 1982.

[25] A.J. Ijspeert, J. Hallam, and D. Willshaw. Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology. *Adaptive Behavior*, 7(2):151–172, 1999.

[26] A.J. Ijspeert and J. Kodjabachian. Evolution and development of a central pattern generator for the swimming of a lamprey. *Artificial Life*, 5(3):247–269, 1999.

[27] A. Crespi and A.J. Ijspeert. 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)*, pages 19–27, 2006.