From Dynamic Hebbian Learning for Oscillators to Adaptive Central Pattern Generators

Ludovic Righetti, Jonas Buchli and Auke Jan Ijspeert
Biologically Inspired Robotics Group
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Switzerland
Email: ludovic.righetti@a3.epfl.ch, jonas.buchli@epfl.ch, auke.ijspeert@epfl.ch

***Abstract*— In this contribution we use a model of adaptive frequency oscillators to build adaptive Central Pattern Generators (CPG). We use a network of adaptive coupled Hopf oscillators to dynamically learn any periodic signal. The signal is then encoded as a stable limit cycle in the network. The interest of this approach is that the learning is not an external optimization process but is embedded in the dynamics of the network. The learning is successful even when the teaching signal is noisy, and the encoded trajectory is stable against perturbations. Furthermore, the learned trajectory can easily be modulated in frequency or amplitude in a smooth way.**

I. Introduction

For the last two decades, models of Central Pattern Generators (CPGs) are increasingly used to control the locomotion of autonomous robots, from humanoids to multi-legged insect-like robots [1]–[4]. CPGs are often modeled by means of coupled nonlinear oscillators [5]. Complex phase patterns can arise from these couplings and therefore, make these systems interesting for modeling gaits of animals and for controlling robots [6]. However, in most cases the design of such CPGs is quite difficult since the different parameters and coupling constants have to be tuned by hand or by an optimization algorithm. Indeed, the values of several parameters usually need to be adjusted, such as the parameters controlling the frequency of the oscillations and their respective phase lags.

In previous contributions [7], we showed that by appropriately converting the parameter controlling the frequency of an oscillator into a dynamical system (i.e. a new state variable in the system), the oscillators were able to adapt their frequency to the frequency of any periodic or pseudo-periodic input signal. Such a mechanism is useful for adapting the frequency of the oscillator to the natural frequency of a mechanical system or to the frequency components of some sensory feedback signals.

Moreover, the form of the adaptive rule we introduced is simple and applicable to different kinds of oscillator. We called this adaptation mechanism Dynamic Hebbian Learning since it is a correlation-based learning rule [8]. The interest of such an adaptive rule is that it is part of the dynamical system, all the learning is embedded in the system and we do not need any supervisor, any external optimization or any preprocessing of the teaching signal. Furthermore, we analytically proved the global convergence of the learning to the desired frequency, for any signal that can be written as a Fourier series. Thus, within this framework, constructing adaptive oscillators based on Hopf, Van der Pol, Rayleigh, Fitzhugh-Nagumo oscillators or Rössler oscillator is straightforward. In other contributions, we already showed how simple adaptive controllers could be built with our adaptive mechanism to control simple robots with spring actuators [9], [10].

In this contribution, we will show how we can extend this general framework to build adaptive CPGs modeled as coupled adaptive oscillators. We show that with such adaptation capabilities, the CPGs are able to learn any desired periodic pattern. The parameters of the CPG are dynamically adapted by the system and no external optimization is required.

We will show that our adaptive CPGs can learn the shape and phase relations of complex periodic inputs. The adaptive CPG learns a periodic input pattern and after convergence, if the input signal disappears, the pattern stays encoded as a structurally stable limit cycle in the system of coupled oscillators. The learning is successful even if the pattern to learn is noisy or if its period is not well-defined. Encoding patterns, or trajectories, as limit cycles is of great interest for controlling robots because the system is robust to external perturbations and can easily integrate sensory inputs. The method we present here can then be used to design robust CPG-based controllers for the locomotion of robots, in particular when an example of the gait is available.

II. Adaptive Central Pattern Generators

In this section, we present our model of adaptive CPGs. We first introduce the idea of dynamic Hebbian learning for oscillators, then we show how the frequency spectrum of a periodic signal can be learned by a network of uncoupled oscillators with a simple feedback loop. These oscillators can learn the frequency and the amplitude of the frequency components of the periodic teaching signal. Finally we introduce coupling between these oscillators in order to keep the correct phase differences between them. This final network corresponds to our model of adaptive CPG.

A. Dynamic Hebbian Learning for Oscillators

Recently, we presented a method for constructing adaptive frequency oscillators [7]. In this method, the parameter influencing the frequency of the oscillator becomes a new state variable and adapts to the frequency of any periodic input

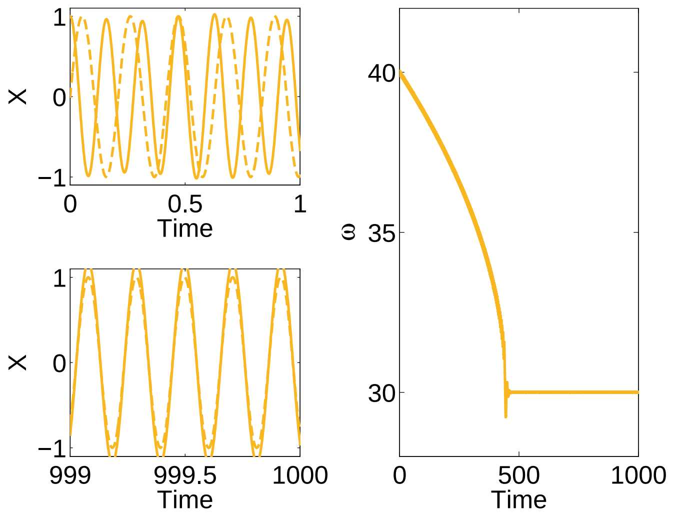

Fig. 1. This figure shows an example of frequency learning in the adaptive Hopf oscillator. The learning input is a simple harmonic signal, $F(t) = \sin(30t)$. The right figure shows the evolution of ω, we clearly see the adaptation to the correct frequency. The left figures show the oscillations of the Hopf oscillator (the x variable), at the beginning of learning (upper graph) and after learning (lower graph), we also plotted the teaching signal F (dashed line). The initial conditions are $\omega(0) = 40$, $x(0) = 1$, $y(0) = 0$, $\mu = 1$ and $\varepsilon = 0.9$.

signal. In this contribution, we use adaptive Hopf oscillators to construct adaptive CPGs. We chose an Hopf oscillator because it possesses an harmonic limit cycle that will be used to decompose the periodic function we want to learn into a sum of sines and cosines. The adaptive Hopf oscillator that we designed is described by the following set of differential equations

$$\dot{x} = \gamma(\mu - r^2)x - \omega y + \varepsilon F(t) \tag{1}$$

$$\dot{y} = \gamma(\mu - r^2)y + \omega x \tag{2}$$

$$\dot{\omega} = -\varepsilon F(t)\frac{y}{r} \tag{3}$$

We can recognize the traditional Hopf oscillator (variables x and y) perturbed by a function F. The frequency of such an oscillator is defined by ω and is then a state variable in the adaptive oscillator. ε is a coupling constant, $\mu > 0$ determines the radius of the limit cycle, γ determines the speed of convergence to the limit cycle and $r = \sqrt{x^2 + y^2}$. In this system, the oscillator will learn the frequency of the periodic input $F(t)$, it means that ω will converge to the frequency of F.

If the input $F(t)$ has several frequency components, the oscillator will generally adapt its frequency to the closest frequency component of the input. It can be shown that for any initial conditions, ω converges to one of the frequency components of F. For a complete discussion about dynamical Hebbian learning, convergence proofs and extension to more complex oscillators see [7]. Figure 1 shows how the adaptive Hopf oscillator learns the frequency of a periodic input signal.

B. Feedback structure and amplitude adaptation

In this section we introduce a network of adaptive Hopf oscillators that learn the frequency components of a periodic input signal and their amplitudes. We introduce a simple feedback loop that controls the learning.

Each oscillator of the network will learn one frequency component of the teaching signal. It will also learn the associated amplitude. When an oscillator has learned the correct frequency and amplitude of a frequency component of the input, this frequency component will then disappear from the teaching signal. In order to achieve this, we use a simple feedback loop described in Figure 2. In this figure, we see that the learned signal is the sum of all the outputs of the oscillators, weighted by the corresponding amplitude. In that sense we have a Fourier decomposition of the input signal. The feedback loop subtracts the already learned signal from the teaching signal. So only the frequency components that were not already learned are still sent to the oscillators.

The network of N oscillators, with frequency and amplitude adaptation, is described by the following set of equations

$$\dot{x}_i = \gamma(\mu - r_i^2)x_i - \omega_i y_i + \varepsilon F(t) \tag{4}$$

$$\dot{y}_i = \gamma(\mu - r_i^2)y_i + \omega_i x_i \tag{5}$$

$$\dot{\omega}_i = -\varepsilon F(t)\frac{y_i}{r_i} \tag{6}$$

$$\dot{\alpha}_i = \eta x_i F(t) \tag{7}$$

$$F(t) = P_{teach}(t) - \sum_{i=0}^{N} \alpha_i x_i \tag{8}$$

where x_i, y_i and ω_i describe the i^{th} adaptive Hopf oscillator. $r_i = \sqrt{x_i^2 + y_i^2}$, η and ε are positive coupling constants controlling the learning rate. P_{teach} represents the input signal to learn. The amplitudes of the frequency components are learned with the α_i variable. The learning rule is a simple correlation based learning (Hebbian type), here α_i increases when the functions $x_i(t)$ and $F(t)$ are correlated, which happens when they have the same frequencies. α_i will stabilize as the associated frequency component will disappear from $F(t)$ when the amplitude is correct.

With the network we presented we are able to learn the frequency components of any periodic input. As long as the frequency spectrum is finite, as each oscillator codes for one frequency component, it is sufficient to have as many oscillators as frequency components in the system. The amplitude of each frequency component is also learned with the α_i variable. For the case of continuous frequency spectrum, we will show in Section III that the network can nevertheless learn a good approximation of the signal. After learning, we set $F(t) = 0$ and the learned signal, $Q_{learned} = \sum_{i=0}^{N} \alpha_i x_i$, is coded in the network of oscillators. But when the learning signal disappears and if perturbations occur, the phase relations between the oscillators will not be kept. In the next section, we add a coupling scheme to the previous network of oscillators. The purpose of such a coupling is to keep the correct phase differences between the oscillators. After learning, the network is composed of uncoupled Hopf oscillators. Each Hopf oscillator

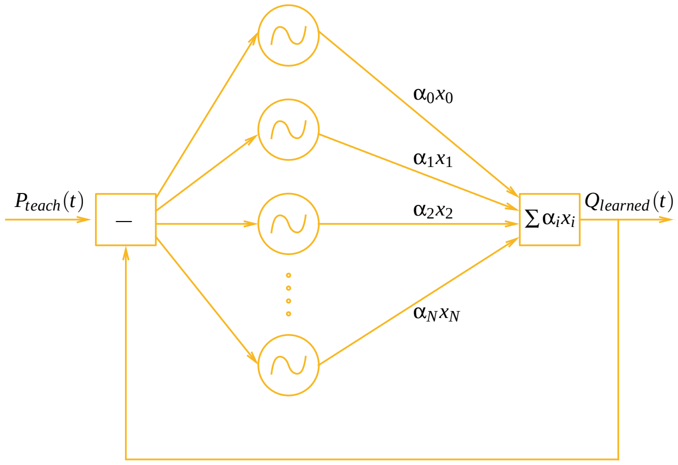

Fig. 2. This figure shows the structure of the network of adaptive Hopf oscillators. Each oscillator receives the same learning signal $F(t) = P_{teach}(t) - \sum_i \alpha_i x_i$, which is the difference between the signal to be learned, $P_{teach}(t)$, and the signal already learned, $Q_{learned}(t)$. Refer to Equations (4)-(8) and to the text for more details.

is structurally stable, so a perturbation on the oscillators will be damped out and the x_i variables of the oscillators will still be sinuses with right frequencies and amplitudes. Nevertheless, the phase relations between the oscillators will be destroyed and thus the $Q_{learned}$ signal will change. We now discuss how to force the correct phase relationships between the oscillators.

C. Stable Central Pattern Generators

Phase relations between oscillators can be achieved by coupling them. Two oscillators, when coupled, are entrained [11] if their frequencies have the relationship of the form $\omega_1 \simeq \frac{p}{q}\omega_2$, $p, q \in \mathbb{N}$. In the case of the Hopf oscillator, entrainment really works only for $p = q$. When numerically constructing the Arnold tongues of a forced oscillator, only the main tongue is clearly visible. Tongues of the form $\frac{p}{q}$ are not visible and thus phase synchronization is difficult or even impossible in numerical experiments.

Nevertheless, we know that it is possible to have 1:1 phase-locking. We can use this fact to derive a coupling scheme for the network. Lets assume that oscillator 0 has the lowest frequency, ω_0, and that this frequency defines also the frequency of the learned signal. Then, as the signal is periodic, we know that its Fourier series will contain frequency components of the form $\omega_i = n\omega_0$, $n \in \mathbb{N}$. The Hopf oscillator has a limit cycle that is a circle in the state space $x - y$. In our case the oscillator is perturbed with signals with small amplitudes, so the limit cycle can still be approximated by a circle, thus we can calculate the instantaneous phase of the oscillator. By using this information and the fact that each oscillators has a frequency that is a multiple of ω_0, we can build a phase signal of frequency ω_i which is in-phase with oscillator 0 by rescaling the phase signal from oscillator 0. This is done in the following equation

$$R_i = \frac{\omega_i}{\omega_0}\,\mathrm{sgn}(x_0)\cos^{-1}\left(-\frac{y_0}{\sqrt{x_0^2+y_0^2}}\right) \qquad (9)$$

Now we have a phase signal R_i which is in-phase with x_0 but has the frequency of oscillator i. If we couple R_i with oscillator i, then we will see phase-locking between oscillator 0 and i. The new equation for oscillator i, with coupling to oscillator 0 for phase-lock, is given by the following equation

$$\dot{x}_i = \gamma(\mu - r_i^2)x_i - \omega_i y_i + \epsilon F(t) + \tau \sin(R_i) \qquad (10)$$
$$\dot{y}_i = \gamma(\mu - r_i^2)y_i + \omega_i x_i \qquad (11)$$

where τ is a coupling constant. Coupling all the oscillators with oscillator 0 assure stable phase-locked oscillations. Nevertheless, we also have to know the phase relations between each oscillator, because it is likely that in a complex signal, the oscillators will not have in-phase oscillations. This phase information must be kept by the system. As we did for frequencies and amplitude, we now introduce a new state variable, to learn the specific phase difference between oscillator 0 and oscillator i, whose equation is

$$\dot{\phi}_i = \sin\left(R_i - \mathrm{sgn}(x_i)\cos^{-1}\left(-\frac{y_i}{\sqrt{x_i^2+y_i^2}}\right) - \phi_i\right) \qquad (12)$$

This equation is a simple first order system where ϕ_i converges to the phase difference between oscillator i and R_i. We use the *sin* function because we are interested in the phase difference modulo 2π.

By using this phase information with the coupling scheme of Equation (9), we can now build a network of oscillators able to keep the correct phase relations between the oscillators. Finally, we have a network of oscillators which is able to learn any periodic input signal that has a first frequency component that defines the frequency of the whole signal (oscillator 0). Each oscillator has 5 state variables, 2 for the oscillatory motion that assure structural stability (x_i and y_i), one for learning the frequency (ω_i), one for the amplitude (α_i) and one for the phase relations (ϕ_i). The network is shown in Figure 3 and summarizes in the following equations

$$\dot{x}_i = \gamma(\mu - r_i^2)x_i - \omega_i y_i + \epsilon F(t) + \tau \sin(R_i - \phi_i) \qquad (13)$$
$$\dot{y}_i = \gamma(\mu - r_i^2)y_i + \omega_i x_i \qquad (14)$$
$$\dot{\omega}_i = -\epsilon F(t)\frac{y_i}{r_i} \qquad (15)$$
$$\dot{\alpha}_i = \eta x_i F(t) \qquad (16)$$
$$\dot{\phi}_0 = 0 \qquad (17)$$
$$\dot{\phi}_i = \sin\left(R_i - \mathrm{sgn}(x_i)\cos^{-1}\left(-\frac{y_i}{r_i}\right) - \phi_i\right), \forall i \neq 0 \qquad (18)$$

with

$$R_i = \frac{\omega_i}{\omega_0}\,\mathrm{sgn}(x_0)\cos^{-1}\left(-\frac{y_0}{\sqrt{x_0^2+y_0^2}}\right) \qquad (19)$$

and

$$F(t) = P_{teach}(t) - \sum_{i=0}^{N} \alpha_i x_i \qquad (20)$$

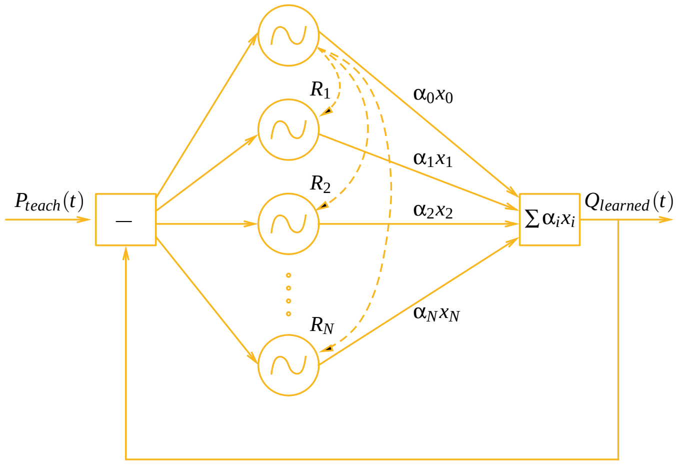

Fig. 3. This figure shows the structure of the network of adaptive Hopf oscillators after we added coupling. Each oscillator receives the same learning signal $F(t) = P_{teach}(t) - \sum_i \alpha_i x_i$, which is the difference between the signal to be learned, $P_{teach}(t)$, and the signal already learned, $Q_{learned}(t)$. Then all the oscillators (except oscillator 0) receive the scaled phase input R_i from oscillator 0. Refer to Equations (13)-(18) and to the text for more details.

We have now presented a network of coupled adaptive Hopf oscillators that can learn a periodic input. We may note that the adaptation is done in a single stage, we do not need to first learn the signal and then add the coupling, the learning takes place in the coupled network of oscillators. In the next section, we will show numerical experiments proving the properties of our adaptive CPG.

III. Experiments

In the previous section, we presented a mathematical model of coupled adaptive oscillators and claimed that it was able to learn periodic signals and encode them in stable limit cycles. We now present numerical experiments to show how our system is working. First we present how it learns a well defined periodic signal. Then we present how learning successfully occurs in the presence of noise. We also show that the system is stable against perturbations. Afterwards, we show that with such a system, we can easily modulate the learned pattern in frequency and amplitude and keep smooth trajectories. Finally, we also show how the system can approximate periodic signals with continuous frequency spectrum.

A. Learning periodic input signals

As an example, we use the system described by Equations (13)-(18) to learn the input signal described by the following equation

$$P_{teach} = 0.8\sin(15t) + \cos(30t) - 1.4\sin(45t) - 0.5\cos(60t) \tag{21}$$

We use a network of 4 oscillators to learn the periodic pattern. The initial frequencies $\omega_i(0)$ are uniformly distributed between 6 and 70. The initial amplitudes $\alpha_i(0)$ and phase $\phi_i(0)$ equal 0.

The results of the experiment can be seen in Figures 4 and 5. An interesting aspect of this learning is that the frequencies

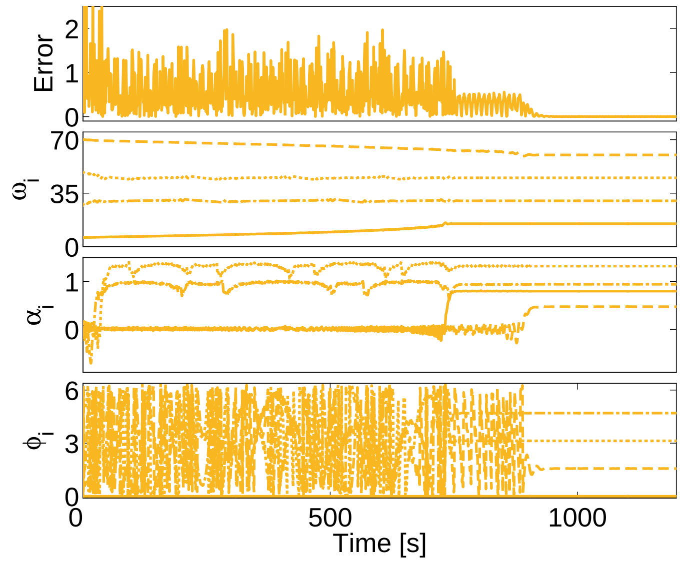

Fig. 4. This figure shows the evolution of the state variables of the network of oscillators during learning of an input signal and the evolution of the error of learning. The signal to learn P_{teach} is defined by equation (21). The upper graph is a plot of the error, defined by $error = \|P_{teach} - Q_{learned}\|$. The 3 other graphs show the evolution of the frequencies, ω_i, the amplitudes, α_i and the phases, ϕ_i. The variables for each oscillator are plotted, variables of oscillator 0 are the plain lines, variables for oscillator 1 are the dotted-dashed lines, variables for oscillator 2 are the dotted lines and the dashed lines represent oscillator 3. The initial conditions are $\alpha_i(0) = \phi_i(0) = 0$, $x_i(0) = 1$, $y_i(0) = 0$ $\forall i$, $\mu = 1$, $\gamma = 8$, $\varepsilon = 0.9$, $\eta = 0.5$ $\tau = 2$. The frequencies $\omega_i(0)$ are uniformly distributed from 6 to 70. Please refer to the text for a discussion of the results.

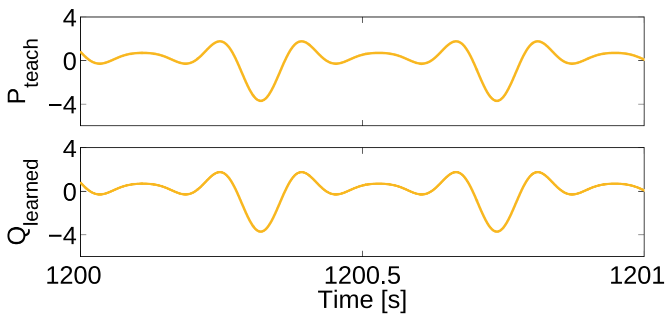

Fig. 5. This figure shows the input signal to learn P_{teach} in the upper graph and the result of learning $Q_{learned}$ in the lower graph. It is obvious that the network correctly learned the input pattern.

of each oscillator are adapted. When a frequency matches a frequency component of the input signal, the corresponding amplitude starts adapting and converges quickly. After oscillator 0 gets the correct frequency, the phase ϕ_i converges to the correct values. By looking at the error plot, we clearly see each time a frequency component is learned. The error finally becomes 0, the signal is completely learned. We see that the frequencies, amplitudes and phase relations are exactly the ones of the teaching signal. Figure 5 shows a plot of equation (21) and of the learned signal, it is obvious that the network of oscillators correctly learned the periodic input.

B. Learning in the presence of noise

In this section we show that the network can learn patterns that are noisy and can even filter out the noise to learn a clean

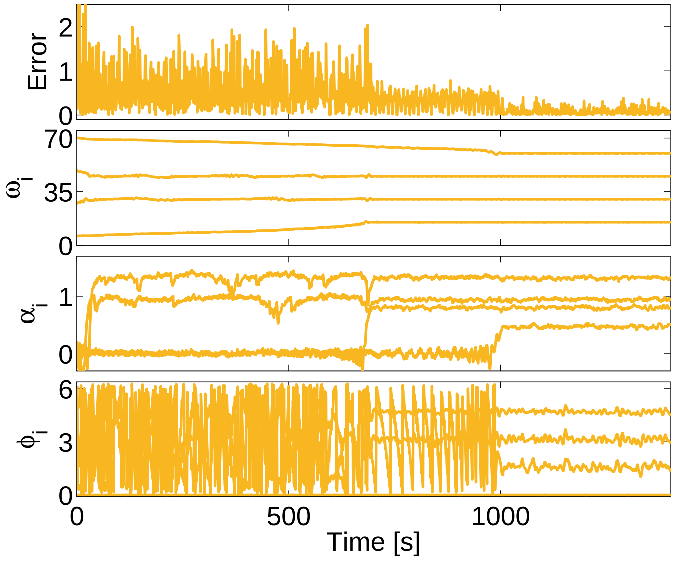

Fig. 6. This figure shows the evolution of the state variables of the network when learning a noisy input. The input is the signal defined by Equation (21) to which we added uniformly distributed random noise between [-2,2]. All the parameters used are the same as in Figure 4. Discussion of convergence is discussed in the text.

pattern.

We take the same learning signal as the previous section and we add uniformly distributed random noise between $[-2,2]$ to this signal. Then we use the same network of oscillators as in the previous section to learn this noisy signal. The results of the experiment can be seen on Figures 6 and 7. These results have to be compared with those of Figures 4 and 5.

First we clearly see that the learned signal corresponds to the teaching signal when the noise is filtered. This phenomena can be explained because the network is learning on a large time scale so what is really learned does not depend on the noise that has a null mean on average and does not change the distribution of the frequencies of the signal.

Nevertheless we see that the convergence is not exact in the sense that the ω_i, α_i and ϕ_i are converging to the right values but are still changing a bit around the correct values because of the noise. This variation is not very visible for the frequencies, but is clear when looking at the amplitudes and the phases. This phenomenon seems evident because the noise is mainly acting on these two parameters.

Finally, even if the convergence is not as exact as previously, the learned pattern corresponds very well to the teaching pattern without noise. In that sense learning is successful.

C. Stability against perturbations

In this section we present a simple experiment to show the stability properties of the oscillator. We take the network we entrained in section III-A and perturb it. To show the strong stability properties, we perturb each oscillator of the network with a random perturbation of intensity 10. It means at time t_p we change the states $x_i(t_p)$ and $y_i(t_p)$ with a random value uniformly distributed between $[-10,10]$.

We notice that such a perturbation is really strong since the limit cycle of the Hopf oscillator is of radius 1, which

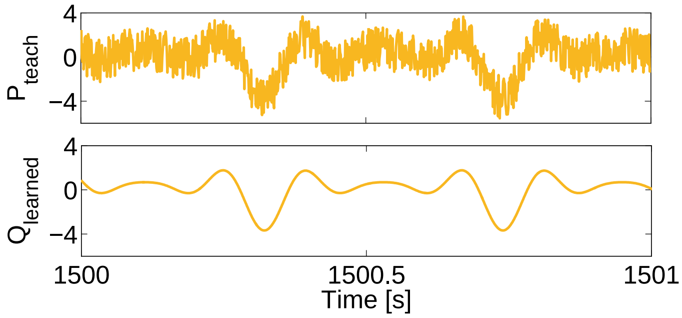

Fig. 7. This figure shows the noisy teaching signal (upper graph) and the learned pattern (lower graph). We clearly see that the signal learned corresponds to the teaching signal from which noise is filtered. A more detailed discussion can be found in the text.

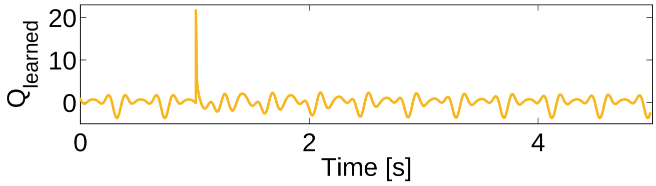

Fig. 8. This figure presents the evolution of the output signal of the network $Q_{learned}$. At time $t_p = 1$ a perturbation occurs on the oscillators of the network. We clearly see that the network quickly recovers its original behavior, thus proving the stability properties of the system. Refer to the text for an extended discussion.

means $x_i, y_i \in [-1,1]$. In Figure 8 we present the result of the experiment. At time $t_p = 1$ we perturb the oscillators, we clearly see on the figure that the resulting perturbation is really strong since $Q_{learned}$, which generally oscillates between $[-3,2]$ goes to 20.

As soon as perturbation is over, the network goes back to its natural region of oscillations, but the shape of $Q_{learned}$ is not exactly the one learned. It takes about 4 cycles to the network to completely recover its original pattern of oscillations. This experiment clearly proves the stability properties of the network of adaptive oscillators.

D. Modulation

In this section, we want to show other properties of the network of oscillators that make it suitable for trajectory generation in robotic applications. After learning, we have 5 vectors of data representing the states of the network, $\vec{x}$, $\vec{y}$, $\vec{\omega}$, $\vec{\alpha}$, $\vec{\phi}$ each vector being in $\mathbb{R}^N$ (N is the number of oscillators). Thus a linear change of the vectors representing the frequency or the amplitude of the trajectory should change smoothly the behavior of the system. For example to generate a trajectory that is 2 times faster, we just have to set $\vec{\omega}$ to $2\vec{\omega}$. A trajectory with a 3 times smaller amplitude is generated by setting $\vec{\alpha}$ to $\frac{1}{3}\vec{\alpha}$. This parameter change does not change the stability properties of the network and we have a smooth transition between the trajectories when the parameters are changed, because of the intrinsic dynamic properties of the system. Such modulations in frequency and amplitude where done and results of the experiments can be seen in Figure 9.

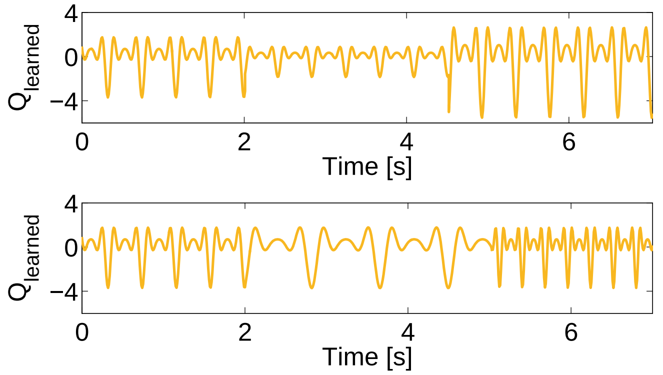

Fig. 9. This figure shows the results of experiments on the modularity of the network. The upper graph shows the behavior of the system when the amplitude is changed. At time $t = 2$, the amplitude is divided by 2 and at time $t = 4.5$ the amplitude is multiplied by 3. The lower graph shows the behavior of the network when the frequency is changed. At time $t = 2$ the frequency is divided by 2 and at time $t = 5$ frequency is multiplied by 3. In both graphs, we can notice the smoothness of the trajectory when the parameters are changed. More details can be found in the text.

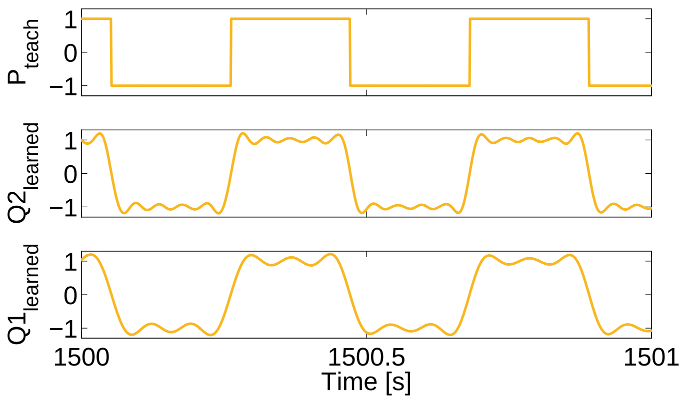

Fig. 10. This figure shows the result of learning of a square signal for 2 networks with different numbers of oscillators. In the upper graph, we see the teaching signal, the middle graph ($Q2_{learned}$) shows the result of learning with 10 oscillators whose initial frequencies are uniformly distributed between $[10, 300]$. The lower graph shows the result of learning with 5 oscillators with initial frequencies uniformly distributed between $[10, 100]$. For the 2 trained networks we set $\varepsilon = 2$, $\tau = 2$, $\gamma = 8$ and $\eta = 0.5$.

E. Learning complex signals

This last section of experiments deals with the case of more complex signals. We showed that our adaptive CPG could encode in a structurally stable limit cycle simple periodic signals. Now we show the case of an infinite spectrum of frequency. In this experiment we use as a teaching signal a square signal, which contains an infinite number of frequency components. We try to learn this signal with the network of oscillators. We made two experiments, with several numbers of oscillators, in order to show that the network can well approximate a signal with an infinite number of frequency components with a finite number of oscillators. In these experiments we also show that the more oscillators we have, the more accurate the learning is. The result of the experiments can be seen in Figure 10.

In these experiments, we clearly see that the network of oscillators can learn a good approximation of the teaching signal with a finite number of oscillators. Furthermore, we note that the higher number of oscillators, the better the learning is.

IV. Conclusion

In this contribution, we showed how we can construct a network of coupled oscillators able to learn periodic signals. We first showed the principle of adaptive frequency oscillators, then we presented a network able to learn the frequency components and the amplitudes of any periodic input signal. The number of frequency components in the teaching signal determines the number of oscillators needed in the network. For the case of continuous frequency spectrum, we showed that it was possible to learn a good approximation of the signal. Furthermore we showed how the correct phase relations between all the oscillators could be kept by an appropriate coupling of the oscillators. The resulting network of oscillators is then stable against perturbations, it means the network has encoded the teaching periodic signal into a limit cycle. The only limitation of this stable network is that stable coupling can be defined only when the lowest frequency of the oscillators also defines the frequency of the signal to learn. Although the solution we presented can learn any periodic input signal, we cannot guarantee the stability of the resulting network of oscillators if the lowest frequency of the oscillators is different from the frequency of the teaching signal.

In a second part, we showed numerical experiments that proved the learning abilities of the network. We showed that even noisy input signals could be learned. Furthermore, we showed that the encoded pattern was stable against perturbations and could be smoothly modulated in amplitude or frequency, making such a network interesting for controlling trajectories in autonomous robots.

Moreover, it becomes easy to learn multi-dimensional periodic signals. We can imagine having a network of coupled adaptive oscillators where the teaching signal for each oscillator would come from several sources and with several output signals. The frequency and amplitude would automatically adapt to learn the multi-dimensional signals and the general coupling structure of the network will assure stability. This ability would be of great interest for controlling multi-DOF robots.

In our future research, we will mainly focus on designing a more general coupling scheme in the network to have stable limit cycles for any periodic teaching signal. Then we will show the applicability of such a network to the control of multi-DOF robots.

Acknowledgment

This work was made possible thanks to a Young Professorship Award to Auke Ijspeert from the Swiss National Science Foundation (A.I. and J.B.) and to the support of the European Commission's Cognition Unit, project no. IST-2004-004370: RobotCub (L.R.).

REFERENCES

[1] G. Taga, “Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment,” *Physica D: Nonlinear Phenomena*, vol. 75, no. 1–3, pp. 190–208, 1994.

[2] ——, “A model of the neuro-musculo-skeletal system for human locomotion. I. Emergence of basic gait,” *Biological Cybernetics*, vol. 73, no. 2, pp. 97–111, 1995.

[3] A. Ijspeert, “A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander,” *Biological Cybernetics*, vol. 85, no. 5, pp. 331–348, 2001.

[4] J. J. Collins and I. Stewart, “Hexapodal gaits and coupled nonlinear oscillator models,” *Biol. Cybern.*, vol. 68, pp. 287–298, 1993.

[5] N. Kopell and G. Ermentrout, “Coupled oscillators and the design of central pattern generators,” *Mathematical Biosciences*, vol. 90, no. 1–2, pp. 87–109, 1988.

[6] M. Golubitsky, I. Stewart, P. Buono, and J. Collins, “A modular network for legged locomotion,” *Physica D*, vol. 115, no. 1–2, pp. 56–72, 1998.

[7] L. Righetti, J. Buchli, and A. Ijspeert, “Dynamic hebbian learning for adaptive frequency oscillators,” *Physica D*, 2005. To appear.

[8] R. Kempter, W. Gerstner, and J. van Hemmen, “Hebbian learning and spiking neurons,” *Physical Review E*, vol. 59, pp. 4498–4514, 1999.

[9] J. Buchli and A. Ijspeert, “A simple, adaptive locomotion toy-system,” in *From Animals to Animats 8. Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior (SAB’04)*, S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J. Meyer, Eds. MIT Press, 2004, conference, pp. 153–162.

[10] J. Buchli, L. Righetti, and A. Ijspeert, “A dynamical systems approach to learning: a frequency-adaptive hopper robot,” in *Proceedings of the VIIIth European Conference on Artificial Life ECAL 2005*, ser. Lecture Notes in Artificial Intelligence. Springer Verlag, 2005, conference, in press.

[11] A. Pikovsky, R. Rosenblum, and J. Kurths, *Synchronization, A universal concept in nonlinear sciences*, ser. Cambridge Nonlinear Science Series. Cambridge University Press, Cambridge, UK, 2001, vol. 12.