IOPscience iopscience.iop.org

Home Search Collections Journals About Contact us My IOPscience

Bio-inspired design strategies for central pattern generator control in modular robotics

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2011 Bioinspir. Biomim. 6 016006

(http://iopscience.iop.org/1748-3190/6/1/016006)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 131.252.130.248
The article was downloaded on 04/02/2013 at 13:24

Please note that terms and conditions apply.

IOP PUBLISHING BIOINSPIRATION & BIOMIMETICS

Bioinsp. Biomim. **6** (2011) 016006 (16pp) doi:10.1088/1748-3182/6/1/016006

Bio-inspired design strategies for central pattern generator control in modular robotics

F Herrero-Carrón, F B Rodríguez and P Varona

Grupo de Neurocomputación Biológica (GNB), Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Calle Francisco Tomás y Valiente, 11, 28049 Madrid, Spain

E-mail: fernando.herrero@uam.es, f.rodriguez@uam.es and pablo.varona@uam.es

Received 2 August 2010
Accepted for publication 17 January 2011
Published 18 February 2011
Online at stacks.iop.org/BB/6/016006

Abstract
New findings in the nervous system of invertebrates have shown how a number of features of central pattern generator (CPG) circuits contribute to the generation of robust flexible rhythms. In this paper we consider recently revealed strategies that living CPGs follow to design CPG control paradigms for modular robots. To illustrate them, we divide the task of designing an example CPG for a modular robot into independent problems. We formulate each problem in a general way and provide a bio-inspired solution for each of them: locomotion information coding, individual module control and inter-module coordination. We analyse the stability of the CPG numerically, and then test it on a real robot. We analyse steady state locomotion and recovery after perturbations. In both cases, the robot is able to autonomously find a stable effective locomotion state. Finally, we discuss how these strategies can result in a more general design approach for CPG-based locomotion.

S Online supplementary data available from stacks.iop.org/BB/6/016006/mmedia

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Effective locomotion is an ability inherent to the animal kingdom. Throughout evolution, life has put to test many different designs to solve the problem. As a result, the present landscape of living forms is a compendium of tested and validated locomotion solutions. Not surprisingly, there are generic mechanisms that solve similar solutions in different contexts: phenotypically distant species use the same strategies to achieve similar goals. Through the study of nervous system commanding locomotion, we can unveil the strategies that can be applied to design novel robotic paradigms.

There are an increasing number of new results on motor control research in living neural systems that remain unexplored in the context of bio-inspired locomotion [1–3]. Of particular interest to robotics are the studies regarding *central pattern generator* (CPG) circuits [4, 5]. CPGs are neural networks that generate rhythmic activity to control motor neurons and are involved in motion that require periodicity, robustness and/or precision. CPGs are autonomous in the sense that they do not need external input to produce a rhythm. However, sensory signals modulate CPG activity in order to adapt to external conditions.

CPGs of invertebrates are the best-known neural circuits in neuroscience research. Recent studies in living CPGs have shown that these circuits (i) have common connectivity building blocks based on mutual inhibition [5–7]; (ii) have neurons and synapses that exhibit rich dynamics with multiple time scales to swiftly negotiate robust sequential activations [8]; (iii) display dynamic invariants to preserve rhythms that are simultaneously robust and flexible [1, 2, 9]; and (iv) have multiple codes that allow cells to multiplex both neural messages and neural signatures, a mechanism that can

 © 2011 IOP Publishing Ltd Printed in the UK

allow a receiver neuron to identify who the sender cell is [10, 11].

In this context, modular robotics provides a flexible platform where different locomotion paradigms can be studied [12, 13]. Using a number of homogeneous modules, one can easily construct different-sized robots, reconfigure their topology or assemble completely newly shaped robots. Moreover, modular robots represent a very good starting point for new paradigms. By their very nature, there are a number of problems that must be solved at different levels of abstraction. For instance, how to code information in one individual module, build a single oscillator or couple oscillators together. Furthermore, modularity calls for generic principles that will scale well when new modules are added in. That is, there is the need for design patterns that can be reproduced locally in each module, while maintaining the global invariant of effective locomotion.

Neuroscientific CPG knowledge has already been successfully applied to robotic control [14, 15] focusing on different aspects: for instance, Ayers *et al* [16] developed a highly realistic motion model of a crustacean limb, while Arena *et al* [17] developed an artificial neural network to control a hexapod robot; different forms of fin/wing control have been achieved by Chung *et al* [18] and Seo *et al* [19] both with an extensive analysis of the convergence and stability of the controllers. Besides these, work has been done on biped locomotion using CPGs [20, 21] and modular locomotion [22], and there have been different approaches to learning, for instance offline genetic CPG design [23] and online optimization methods [24, 25].

In most cases, CPG bio-inspiration in robotics uses the scientific knowledge from these circuits that was available more than 20 years ago. Thus, bio-inspiration is often reduced to the use of oscillators implemented with basic single time-scale limit cycle behaviour. While this type of CPG control has proved highly successful, in this paper we argue that the use of novel findings regarding living CPGs can result in more general design strategies for autonomous locomotion in modular robots. We argue that the proposed biological strategies will provide greater flexibility and robustness and lead to more autonomous behaviour.

In this paper, we first introduce relevant results of recent CPG research; we then apply some of the studied strategies to illustrate the control design of a simple modular robot. After this we study the stability of the generated signals: amplitude, frequency and synchronization, and finally we present the results of the robot in the real world. An appendix illustrates the implementation of a feedback mechanism between a servo and our CPG to achieve entrainment between them.

2. Recent results on living CPG research

In order to build effective artificial CPGs, we want to study what common strategies living CPGs follow for their own work. One ubiquitous feature of invertebrate CPGs is that they are built with non-open topologies, i.e. every neuron receives at least one connection from another CPG member. This is called a 'non-open' network topology, in contrast to an 'open' topology, where at least one neuron does not receive synapses from any other CPG member [6, 7]. In principle, one could build CPGs by having one single pacemaker neuron drive a set of other neurons. However, even CPGs containing pacemakers provide some feedback to it from the rest of the circuit [2, 5].

Together with non-open connection architecture, the ability of individual elements of the circuit to operate on multiple time scales allows CPGs to produce signals that take into account present and past status of the circuit rhythm. During non-transient behaviours, CPG circuits produce robust rhythms that seem to be built with low-dimensional oscillators [8]. However, individual CPG neurons, when isolated, display a spiking–bursting activity capable of generating highly irregular rhythms [5]. This rich dynamics is built with slow and fast time scales, which are also present in the synaptic connections. We thus believe that in order to achieve richer rhythms and build CPGs that can perform complex tasks, we need to take advantage of multiple time-scale mechanisms.

It is important to emphasize that living CPG rhythms are not only robust; they are also flexible and capable of undergoing fast transients in which neurons negotiate their dynamics to create new rhythms from external inputs. Non-open CPG topologies are mainly based on mutual inhibition, which combined with the rich behaviour of individual neurons generates a so-called *winnerless competition dynamics* among them [8]. It is the negotiating nature of this dynamics that endows CPG circuits the ability to build a wide variety of motor commands required for autonomous locomotion.

First we will introduce the basic elements we propose to build CPGs: neurons and synapses, and show how modular robotics can take advantage of a set of design principles derived from the discussed phenomena observed in living CPGs.

3. Bio-inspired CPG components

Most of these strategies, all the more so those regarding architecture, are independent of the individual details of neurons and synapses. However, for our purpose, we demand specific characteristics of a neuron model and a synapse model. Specifically, the neuron model of choice must be able to robustly encode locomotion information, while at the same time being flexible enough to negotiate a stable rhythm, and the synapse model must be able to process different time scales in the neuron model.

3.1. Multiple time-scale neuron model

CPG neuron models for robot locomotion typically use oscillator models with one single time scale. The negotiation capacity of these units to produce robust yet flexible rhythms within the network is more limited than that of neurons with multiple time scales. Multiple time scales can account for a wider variety of bifurcations and transient dynamics to provide autonomous coordinated responses [5, 8]. Indeed, multiple time-scale behaviours are ubiquitous in real living CPG neurons [5].

In our work, we use a neuron model that mimics the activity of real multiple time-scale neurons. Developed by

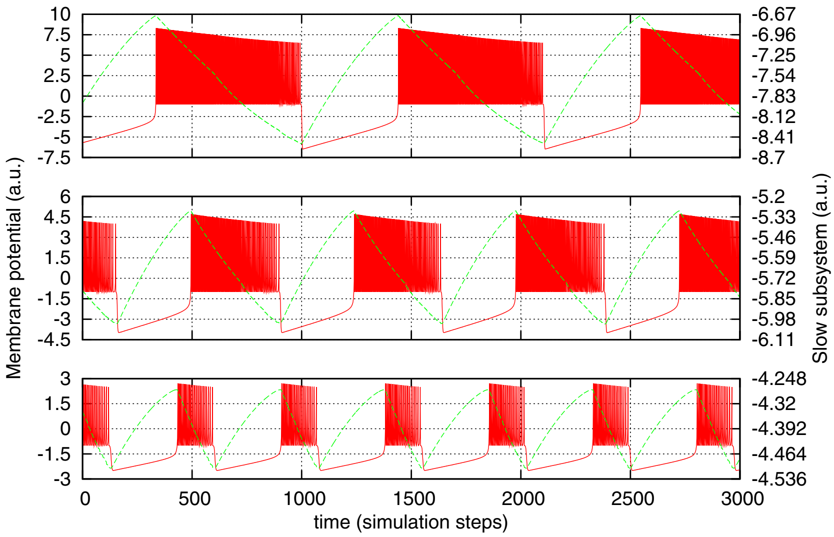


Figure 1. Fast and slow subsystems of Rulkov's neuron model. The slow subsystem (y_n in (1*c*)) is responsible for signalling the beginning and end of a burst; the fast subsystem (x_n in (1*b*)) is responsible for the oscillations that generate individual spikes within each burst. Bottom panel: $\alpha = 7$ and $\sigma = -0.33$; centre panel: $\alpha = 10$ and $\sigma = 0$; top panel: $\alpha = 15$ and $\sigma = 0.33$.

Rulkov *et al* [26, 27], several characteristics have played in its favour: its mathematical simplicity and the possibility to easily control the possible set of behaviours depending on the selection of a few parameters. Three stable regimes may be selected by combination of its parameters: silent, in which the potential of the neuron (variable x_n in (1*b*)) remains in a constant resting state; tonic spiking, in which the neuron emits spikes at a constant rate; and tonic bursting, in which bursts of spikes are emitted at a constant rate, with a silent interval in between. Furthermore, in the boundaries of the parametric regions of those regimes, chaotic behaviour may be found [27]. Of these behaviours, tonic bursting is the one of greatest interest to us. See figure 1 for an overall idea of the model working in the tonic bursting regime.

The bursting regime of the model presents a slow wave (slow time scale) with fast spikes of activity sitting on top of it (fast time scale). We use the slow time scale (y_n in (1*b*)) to encode movement duration, i.e. the temporal length of the burst defines the temporal length of the movement, and the fast time scale (x_n in (1*b*)) to define angular velocity of the servo (higher frequency of the spikes corresponds to faster servo movements).

The mathematical description of Rulkov's model as used in this work is as follows:

$$f(x,y) = \begin{cases} \dfrac{\alpha}{1-x} + y & \text{if } x \leqslant 0 \\ \alpha + y & \text{if } 0 \leqslant x < \alpha + y \\ -1 & \text{otherwise} \end{cases} \tag{1a}$$

$$x_{n+1} = f(x_n, y_n) \tag{1b}$$

$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma + \mu I_n \tag{1c}$$

with $\mu = 0.001$ in all experiments.

This is a bi-dimensional model, where variable x_n represents a neuron's membrane voltage and y_n is a slow dynamics variable with no direct biological meaning, but with similar meaning as *gating* variables in biological models that represent the fraction of open ion-channels in the cell. While x_n oscillates on a fast time scale, representing individual spikes of the neuron, y_n keeps track of the bursting cycle, a sort of context memory. Units are dimensionless and can be rescaled to match the requirements of the robot.

The combination of σ and α selects the working regime of the model: silent, tonic spiking or tonic bursting. In the bursting regime, these parameters also control several properties of neural activity. Figure 4 shows the relationship between parameters α and σ and several properties of the neuron. For instance, the period of the neuron depends almost linearly with α, so the larger its value, the larger the period. These two parameters may be used to tune the locomotion of the whole CPG.

Finally, the external input is modelled through I_n. Depending on this value, a neuron will modify its behaviour. For instance, an external driving force may be input using this parameter. This property is essential for autonomous organization: processing units in the CPG must be able to negotiate the rhythm among them. Also entrainment between the CPG and the physical robot can be achieved through I_n by adding an error term as the external input to a neuron (see the appendix for an example implementation). The total effect of this parameter will depend upon past history of events, the exact value of I_n and the phase within the burst cycle at which the neuron finds itself. In our work, I_n is the current flowing from one neuron to another: a periodic sampling of the continuous function described below in (3).

3.2. *Kinetic synapse model for interneuron communication*

A key property of CPGs is that they are autonomous, i.e. the different units in the circuit talk to each other to negotiate the overall function. Here we present the model we have chosen to implement synapses, the communication channel of neurons. In this work we use a chemical synapse model [30].

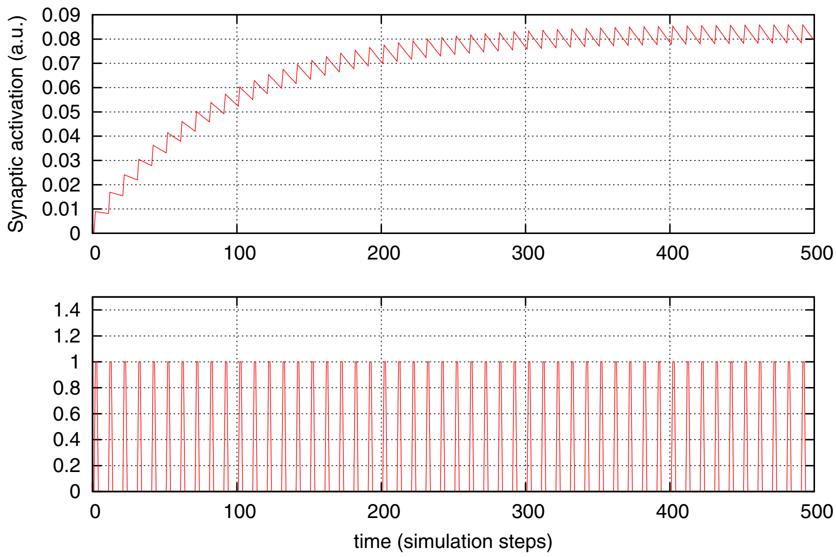


Figure 2. Synaptic response (upper panel) to a train of spikes arriving from the presynaptic neuron (lower panel). For each spike from the presynaptic neuron, a small amount of neurotransmitter is released in the synapse that binds to receptors in the postsynaptic neuron. After a short time, transmitters begin to unbind from receptors. The process of binding and unbinding causes the characteristic sawtooth shape. This model of synapse shows a memory effect, in the sense that response to any given pulse depends on past history of events. The variable r of the model (upper panel) represents the rate of bound receptors in the postsynaptic neuron. According to (3), the current that will flow into the postsynaptic neuron is proportional to the rate of bound receptors. That is, when a neuron emits a series of spikes, the current that will flow into the postsynaptic neuron has a dynamic behaviour, rising with time and converging to a stable value.

Chemical synapses are unidirectional. When a potential spike arrives from the presynaptic neuron, the synapse releases a certain amount of neurotransmitter molecules that bind to the postsynaptic neuron's receptors. With time, neurotransmitter molecules begin to *unbind*. If a succession of spikes arrives within a short time, the synaptic response to each of them may overlap. Therefore, the state of the synapse is dependent upon past events, a mechanism of context memory (see figure 2).

The additional time scale provided by kinetic synapses in a CPG enriches synchronization between bursting neurons. For instance, we may choose to synchronize two bursting neurons upon the spike (fast) time scale or the burst (slow) time scale. We have selected the kinetics of the binding and unbinding processes such that synapses act as filters of the fast time scale and synchronization occurs at the slow time scale. That is, the basic unit of synchronization will be the burst as a whole, not every individual spike. Beyond this, synapses may introduce delays for a finer control of phase difference between neurons.

The mathematical description of the model follows:

$$\dot{r} = \begin{cases} \lambda[T](1-r) - \beta r & \text{if } t_{\mathrm{f}} < t < t_{\mathrm{f}} + t_{\mathrm{r}} \\ -\beta r, & \text{otherwise} \end{cases} \qquad (2)$$

This equation defines the ratio of bound chemical receptors in the postsynaptic neuron (see figure 2 for a sample trace), where r is the fraction of bound receptors, λ and β are the forward and backward rate constants for transmitter binding and $[T]$ is the neurotransmitter concentration. The equation is defined piecewise, depending on the specific times when the presynaptic neuron fires (t_{f}): during t_{r} units of time, the synapse is considered to be releasing neurotransmitters that bind to the postsynaptic neuron. After the release period, no more neurotransmitter is released and the only active process is that of unbinding, as described by the second part of the equation. Times t_{f} are determined as the times when the presynaptic neuron's membrane potential crosses a given threshold θ.

Synaptic current is then calculated as follows:

$$I(t) = g \cdot r(t) \cdot (X_{\mathrm{post}}(t) - E_{\mathrm{syn}}) \qquad (3)$$

where $I(t)$ is postsynaptic current at time t, g is synaptic conductance, $r(t)$ is the fraction of bound receptors at time t, $X_{\mathrm{post}}(t)$ is the postsynaptic neuron's membrane potential and E_{syn} its reversal potential, the potential at which the net ionic flow through the membrane is zero. When coupling two Rulkov map neurons we will need to use a discrete synaptic function. We will build a sequence, let us call it I_n, by simulating $I(t)$ as a continuous function and then taking samples every 0.01 time units (for our choice of kinetic parameters as outlined in the different figures).

We say that a synapse is excitatory when the probability of the postsynaptic neuron firing a spike increases after the presynaptic neuron has fired. If the probability decreases, the synapse is inhibitory. If the postsynaptic neuron rhythmically emits spikes, an excitatory synapse will generally increase its frequency while an inhibitory one will generally decrease it.

4. Bio-inspired strategies for the design of a modular CPG

In this section, we will use an example to illustrate the process by which we have designed a bio-inspired CPG to control a modular worm robot[1] by González *et al* [31]. This platform is very powerful, in terms of locomotion capabilities, while still

[1] More information about the robot is available at http://www.iearobotics.com/personal/juan/doctorado/cube-revolutions/.

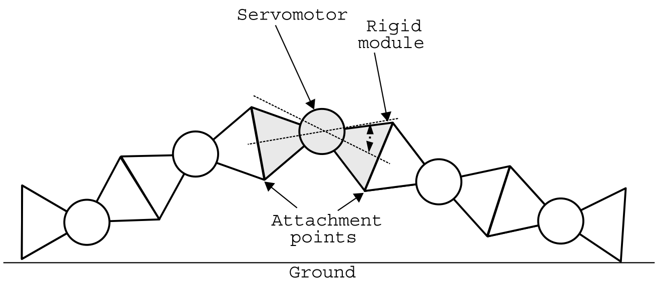


Figure 3. General schema of the worm robot. A single module is marked in grey; all other modules are exactly equal to this one.

being very accessible and easy to control. For simplicity, we have focused on horizontal ground displacement, one of the many locomotion modes this robot is capable of.

The robot, illustrated diagrammatically in figure 3, consists of several modules attached side by side through special connection points. Each of these modules consists of two triangle-shaped rigid pieces, joint by one vertex of the triangle, and a servomotor controlling the angle between these two pieces. In the horizontal locomotion mode, modules are connected sequentially, each of them oscillating on the same plane. One solution to the control problem posed here is undulatory locomotion. Each module must oscillate periodically at a given phase lag from the neighbouring ones. Thus, the CPG must solve the problem of individual oscillation and global coordination.

The choice of this platform has been motivated by its versatility (the reader is again referred to [31]), low cost and ease of construction. The chassis is built of methacrylate panels, assembled by hand in less than 1 h. The servos are Futaba S3003, readily available in any RC store and with an approximate cost of $15 a piece. Finally, there being no wheels, limbs or any other movable parts besides the servos, the control of the robot is exclusively a problem of synchronization among modules, a problem that CPG control will solve in a robust and flexible manner.

As a summary of the section, we follow a bottom-up approach: first, we devise how to code locomotion commands using a neuron model with multiple time scales; then we build a neural circuit based on a non-open topology and mutual inhibition that will make one single module oscillate; finally, we provide modules with a communication channel with their neighbours for them to communicate and negotiate with each other, following the non-open topology strategy, which effectively gives rise to winnerless competition dynamics by guiding the coordination of the activity of multiple time-scale neurons.

4.1. Locomotion information coding: exploiting multiple time scales

The goal of a CPG controller is to generate motor signals that will ultimately drive a motor plant. Such signals must be coordinated among themselves and they must carry appropriate information for locomotion to be effective. In our particular case, the targets are servomotors with only one degree of freedom. A locomotion command for one servo needs three parameters: duration of the movement, velocity and direction. Therefore, we need a mechanism to encode this information in neural activity.

A key biological strategy that we take advantage of is the ability of single neurons to handle multiple time scales. With this mechanism we can multiplex the different aspects of one locomotion command in a single variable, namely membrane potential, in a very robust way [32]. Using one bursting neuron, each individual spike will trigger an atomic action; the number of spikes in one burst will encode the amplitude of movement; and the frequency of the spikes will define the speed of movement. In this way, we can use the neuron model exposed in the previous section to drive the servomotors in our robot. Through control of the parameters of the model, α and σ, properties of the bursts (see figure 1) may be controlled as seen in figure 4. For instance, the length of one period can be controlled almost linearly by tuning the α parameter. Also the ratio of spiking activity to silent activity within one period is better controlled by adjusting the σ parameter.

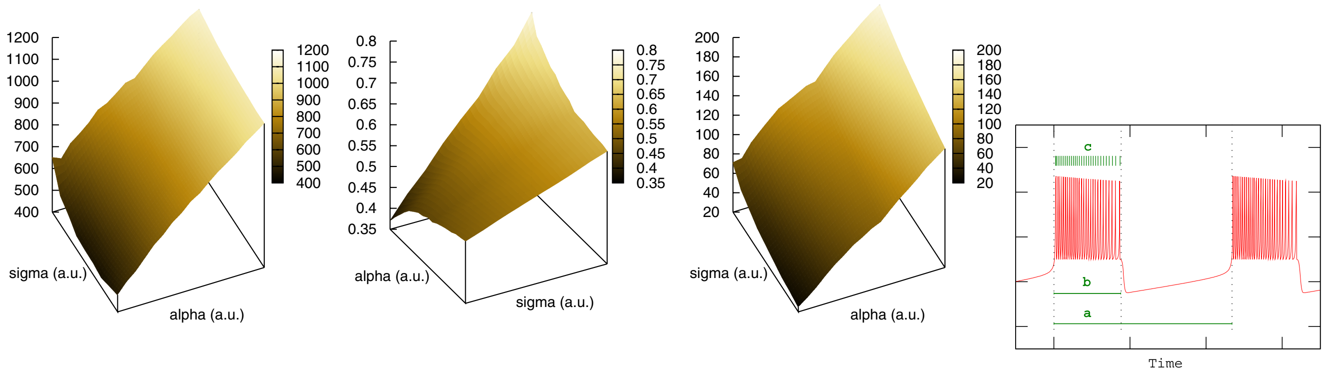
It is worth noting that this ability is independent of the specifics of the neuron model used, as long as it has the necessary property of bursting activity. The model that we have chosen is capable of robustly encoding locomotion information even in the presence of noise.

4.2. Building a CPG to drive one single module: non-open topologies and mutual inhibition

We are designing a CPG to control a robot made of a number of homogeneous individual modules. Since the robot can be reconfigured by adding or subtracting modules, ideally the reconfiguration of the CPG should be as easy as that of the actual hardware. Therefore, we have chosen to first design a small CPG that will drive one single module, and then implement inter-module negotiation mechanisms. Here we will explain how information encoded in neurons' activity is decoded and how the signal that drives the motors is generated.

We begin with the design of a CPG that will drive one single module (servo). This same architecture is repeated in every module. We use a strategy found in many species for the generation of alternate rhythms, where neurons controlling one side of the body inhibit antagonist neurons on the other side of the body [29]. This strategy, named *half-centre oscillator*, subsumes three of the basic strategies that we have outlined: non-open topologies, so that every member of the CPG has knowledge of the overall working of the system, mutual inhibition and, as a result, winnerless competition, in order to generate a reproducible sequential activation.

The need for anti-phase synchronization arises naturally since for a rhythmic activity to be produced, there must be a sequence of 'doing', 'undoing', 'redoing' and so on. Take for instance walking. Each limb continuously repeats a cycle of stance and displacement. Lifting the limb from the ground is actively performed by a certain group of muscles; placing the limb back on the ground is usually a collaboration between relaxation of the previous group, gravity and active



(*a*) (*b*) (*c*) (*d*)

Figure 4. Different properties of an isolated neuron (Rulkov's model) in the bursting regime for different values of α (from 6 to 16) and σ (from −5 to 5). Controlling these parameters, the global locomotion of the robot can also be adjusted. For instance, parameter α has an almost linear relationship with any of the properties analysed here. Thirty consecutive bursts in a stable regime were analysed: *(a)* mean period, measured in simulation steps; *(b)* mean duty cycle, measured as the percentage of the period that corresponds to spiking activity; *(c)* mean number of spikes per burst of 30 consecutive bursts. *(d)* Explanation of magnitudes: a is the period, b/a is the duty cycle and c is the number of spikes per burst.

performance of an agonist group of muscles. For the cycle to be effective the two agonist groups must be activated in a non-overlapping sequence, or anti-phase synchronized.

The architecture of one single module is shown in figure 5(*a*). Two endogenously rhythmic neurons (R and P) are interconnected with inhibitory synapses. The role of inhibition is to prevent both neurons from firing at the same time: when one fires, the other's activity is delayed; in turn, when the second neuron is bursting, it delays the first neuron's activity and the cycle begins again. It is worth noting that bursting neurons are flexible enough to negotiate a global rhythm while still being able to independently code locomotion information. Thus, after a process of synchronization, in which each neuron is capable of encoding their own information, they both arrive at a steady anti-phase state (figure 5(*b*)). Entrainment between the physical module and the neurons is possible through parameter I_n in (1*c*).

4.3. Translating from neural code to motor actuator commands: motoneurons

Movement information is robustly encoded in the neurons' bursting episodes. A neuron called *motoneuron* is then responsible for *decoding* this information and translating it into the signal that will finally be sent to the servo controller. This signal tells the angle at which the servo should be positioned, in degrees. Figure 5(*b*) shows an example pattern of activity of an isolated module in its steady state, after an initial transient period of self-adjustment.

In [33], biological evidence that muscles actually summate spikes of a burst is provided (see also [34, 35]). Muscles that act in this way achieve a contraction state, which they call the 'tonic component' of the neural command, and then rhythmically show small contractions provoked by each individual spike. This behaviour is very similar to the motoneuron model that we will introduce in this section.

With the motoneuron model provided in our paper, we have tried to mimic the real transformation occurring between living motoneurons and muscles. However, a key issue in the design of biomimetic devices is that the principles underlying activation of natural joints and artificial motors are qualitatively distinct. Nonetheless, the departure from pure biological inspiration is not as far as it may seem. In any case, there needs to be a nonlinear process that ultimately translates from neural activity to a physical action.

Motoneurons *read* the activity of the modular oscillator through a pair of synapses. These synapses connect R and P neurons to the motoneuron and are governed by a very simple threshold equation

$$s(x, \nu) = \begin{cases} 1 & \text{if } x > \nu \\ 0 & \text{otherwise.} \end{cases} \qquad (4)$$

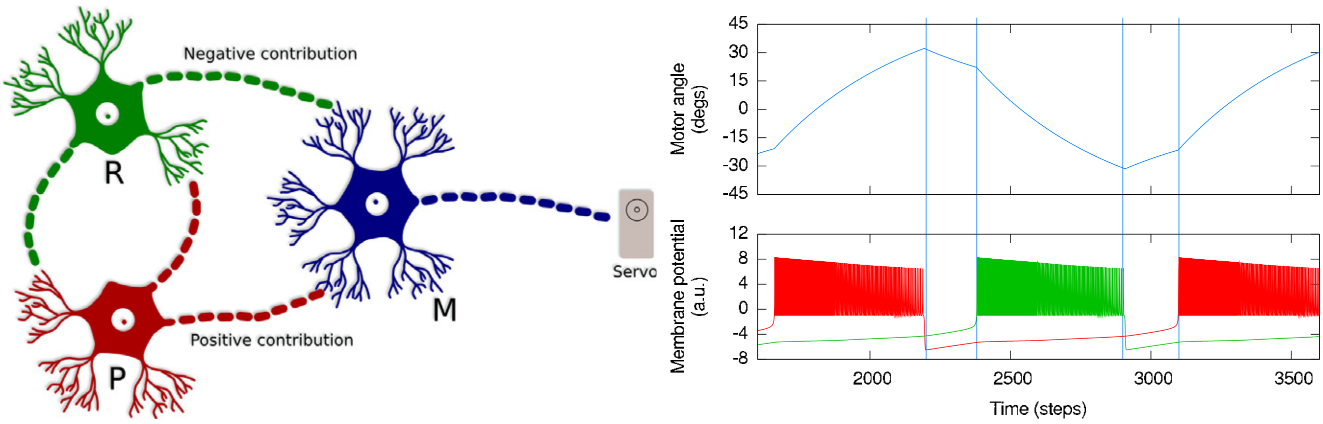
The role of this function is to detect individual spikes of neurons. By setting the threshold to, for example, $\nu = -1.5$ au, this function applied to the potential trace of one neuron will have value 1 during individual spikes and 0 otherwise. In this way, communication between neurons is event-based. That is, the actual shape of neural activity is not so important, only their timing is. We believe this is a mechanism that the nervous system employs to lower the impact of noise [32].

The role of motoneuron M is now to integrate the individual events emitted by each one of the neurons. If neuron P emits a spike, motoneuron M will move the servo a little bit in a positive angle. If it emits a second spike close enough to the first one, the servo will be positioned a little bit further. Analogously, the R neuron will make the motoneuron move the servo towards negative angle positions. If both neurons are silent, motoneuron M will slowly drive the servo to a resting position of angle 0. This is accomplished through the following equation governing motoneurons in our CPGs:

$$C(t) = \gamma[s(x_{\mathrm{p}}(t), \nu) - s(x_{\mathrm{r}}(t), \nu)] \qquad (5)$$

$$\tau\dot{m} = C(t) - m(t) + \text{offset}, \qquad (6)$$

where $m(t)$ is the output of neuron M (in degrees), the two $s(.)$ terms are the threshold function (4) applied to input from R



(*a*) (*b*)

Figure 5. *(a)* Organization of the CPG within one module. The promotor (P) and remotor (R) neurons are interconnected with inhibitory synapses so that they synchronize in anti-phase. The motoneuron M sends a command signal to the servomotor specifying the angle at which the servo should position itself. The signal generated by R is directly input to M, and the opposite of the signal generated by P is input to M. M integrates its input according to (6). *(b)* Activity sample (variable x_n in (1*b*)). Neuron P contributes positively and raises M to 30°; neuron 'R' does exactly the opposite and drives M towards $-30°$. P and R are synchronized in anti-phase. With no input, M tends to 0°. Parameters for P and R: $\alpha = 15$, $\mu = 0.001$, $\sigma = -0.33$, $\beta_e = 0$, $\sigma_e = 1$; parameters for the inhibitory synapses between P and R: $\lambda = 0.5$, $\beta = 10$, $\tau = 0$, $E_{syn} = 9$, $g_{syn} = 1.5$, $T = 1$, $t_r = 0.01$; parameters for M: $\gamma = 30$, $\tau = 0.5$, $\nu = -1.5$.

and P, τ is a time constant that controls how quick the output signal $m(t)$ will change and γ defines the maximum amplitude of signal $m(t)$. The parameter *offset* will add an offset so that the servo oscillates around that value instead of zero. In all results of this paper, *offset = 0*.

In this equation, P contributes positively and R negatively. Given the fact that P and R oscillate in anti-phase, the solution $m(t)$ is an oscillatory function bounded between $-\gamma$ and γ.

When the motoneuron receives no input because P and R are silent, it will go back to zero due to the leak term $(-m(t))$ in (6) (see the decay between bursts in figure 5(*b*)).

4.4. *Inter-module communication to generate a reproducible activation sequence: winnerless competition dynamics*

The final step in implementing our CPG is defining the restrictions that will govern the working of the CPG in its search of a stable rhythm. That is, we want to *program* the collective behaviour of all the neurons within the network. We call this approach *dynamical invariant programming* [36].

Here, we are interested in implementing a particular kind of network dynamics named *winnerless competition* [8]. In this type of dynamics, all neurons *compete* with each other through inhibition. When one neuron is active, it will inhibit some other neurons, preventing them from activating as well. The key point is that there must be a mechanism by which this inhibition is released. When this occurs, the previously inhibited neurons are allowed to become active, inhibiting other neurons in turn. With this release mechanism, it is ensured that no single neuron will inhibit all other neurons permanently, hence the term 'winnerless' competition. Beyond this, we seek a mechanism to implement a winnerless competition in which the sequence of activation is reproducible, in order for the robot to undulate properly.

In summary, three principles are required for generic winnerless competition dynamics: non-open topologies, asymmetric inhibition and a mechanism by which inhibition is released, guaranteeing that no neuron will be permanently inhibited. Together with these, we consider that the topology of the CPG must be modular, as that of the robot, and that the sequence of activation must be reproducible. With this in mind, we proceed with the design of the inter-module architecture. We have come up with different possible designs based on these assumptions. We will describe a particular case here, illustrated in figure 7. This solution is relatively independent of the details of each single module. That is, winnerless competition and the architecture proposed here is a mechanism by which elements produce a reproducible sequence, irrespective of whether their activity is bursting or spiking.

We have added two bistable neurons (see (7*b*) through (7*d*)), which are a modified Rulkov map (see (1*a*) through (1*c*)) to every module. Their role is to inhibit the promotor and the remotor neurons respectively to impose an ordered activation of the modules. They can be either silent or in a tonic spiking regime: when they are excited they switch to the tonic spiking regime until an inhibition occurs; when they are silent, they will remain in that state as long as there is no excitation. That is, upon receiving excitation, the promotor (remotor) neuron will be inhibited *until* an inhibitory signal is received. Once this occurs, the promotor (remotor) neuron is free to burst or remain silent *until* an excitatory signal is received:

$$f(x, y) = (1a) \tag{7a}$$

$$x_{n+1} = f(x_n, y_n) \tag{7b}$$

$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma_n \tag{7c}$$

$$\sigma_n = \begin{cases} 0.33 & \text{Whenever } I_n > 0 \text{ until } I_n < 0 \\ -0.33 & \text{Whenever } I_n < 0 \text{ until } I_n > 0. \end{cases} \tag{7d}$$

The sources of inhibition and excitation are, respectively, the following and preceding modules. If the promotor of module $n-1$ is active, the promotor neuron of module n will be inhibited until the promotor neuron of module $n+1$ begins bursting. When this occurs, the promotor n will be released from inhibition and will be free to fire. When it does, module $n+1$ will be inhibited, and so on. This leads to the desired *winnerless competition*.

Border neurons would only receive signals from one side, not from both. If they were left to burst freely, they would do so at a higher frequency than the rest, since they would be receiving no inhibition. Thus the need for a non-open topology emerges naturally. That is, border neurons need to receive some feedback from the rest of the CPG. Adding a 'border synapse' regularizes the CPG and a stable rhythm may be achieved. The following section will further discuss the effects of having an open and a non-open topology.

5. Analysis and quantitative results

In this section we will analyse a concrete CPG model and study the stability of various attributes of the generated motor signals, namely their amplitude, instantaneous frequency and phase difference.

In order to analyse one signal generated for one motor by the CPGs, we first generate an analytic signal that uniquely represents it. The basic tool for our analysis is the Hilbert transform (a complete revision of the theory can be found in [37]). Basically, the Hilbert transform of a real-valued function/sequence is another real-valued function/sequence whose Fourier components are shifted 90° with respect to the original.

Let $m = \{m_1, m_2, \ldots, m_n\}$ be a signal generated for one of the motors (thus a real-valued sequence). We will denote $\mathcal{H}\{m\}$ the Hilbert transform of the sequence, which is also real-valued and $\mathcal{H}\{m\}_i$ the ith element of the resulting transformed sequence. The equivalent analytic signal representation of m is

$$Z = \{Z_i = m_i + j \cdot \mathcal{H}\{m\}_i\}. \tag{8}$$

This representation allows us to model Z as an amplitude and phase-modulated oscillator

$$Z_i = A_i \exp(j\phi_i) \tag{9}$$

such that

$$A_i = |Z_i| = \sqrt{m_i^2 + \mathcal{H}\{m\}_i^2} \tag{10}$$

and

$$\phi_i = \angle Z_i = \arctan\left(\frac{\mathcal{H}\{m\}_i}{m_i}\right). \tag{11}$$

Furthermore, we can now reconstruct the original signal m as

$$m_i = A_i \cos(\phi_i). \tag{12}$$

There has been extensive discussion about the applicability of this technique in general cases, particularly when calculating the instantaneous frequency derived from instantaneous phase [38]. The two main restrictions applicable to signals for their instantaneous frequencies to make physical sense are that they be mono-component and that they oscillate symmetrically around zero. Generally speaking, a mono-component signal is one which does not have sub-oscillations between zero-crossings, i.e. there is only one local extreme between zero-crossings.

In order to gain some insight as to how the analytic signal is related to the original signal, we present two examples in figure 6. In this figure we show two signals and the complex plane projection of their corresponding analytic signals. The first signal is a mono-component signal. Its analytic signal projected on the complex plane shows an oscillatory orbit around the origin. Phase calculated as in (11) will yield a monotonically increasing value within each period of the signal. The other case is a non-monocomponent signal. In this case, since the projected signal does not describe a simple orbit but displays some loops, there will be at least two distinct points within one period that will have equal phase value.

Signals of our CPGs have been filtered before plotting with a moving average filter. The window size is 1000 points of width, a little less than the mean period of the signals. This way we eliminate the small noise in the plot and keep the general behaviour. Noise in the original signal does not result in jerky locomotion of the robot as can be seen in the online supplementary videos, available from stacks.iop.org/BB/6/016006/mmedia.

5.1. Case study: non-open topology

We will apply the analytic signals' technique to analyse three parameters of the signals that command the motors of our robot, namely amplitude, frequency and phase difference among adjacent modules. We first study the behaviour of the CPG depicted in figure 7.

5.1.1. Amplitude. Since our CPG is operating a real-world robot, one major concern is that the output be kept stable and within a desired range that will cause no harm to it. We want to guarantee that this example CPG is capable of autonomously constraining the amplitude of the output signal.

The amplitude envelope of a given signal m can be calculated from the corresponding Z as in (10). It is important to note that this function yields a value for every time step i, i.e. we can define an amplitude value even for sequence members of m where the signal is not at an extreme. Figure 8(*b*) shows the amplitude envelope of the signal generated by the bistable CPG in figure 7 for motor 8 of the robot. There is a transient period at the beginning of the simulation during which neurons P and R within the module are not yet synchronized. The amplitude of oscillation is low during this transient. Once the system stabilizes, oscillation reaches its nominal amplitude, which is kept constant for the rest of the simulation.

5.1.2. Phase difference. Efficient locomotion of the robot is achieved when adjacent modules maintain a constant phase difference between them in the steady state. In order to study phase differences, we take signals $m^{(k)}$ and $m^{(l)}$, $k \neq l$, corresponding to motors k and l and construct $Z^{(k)}$ and $Z^{(l)}$.

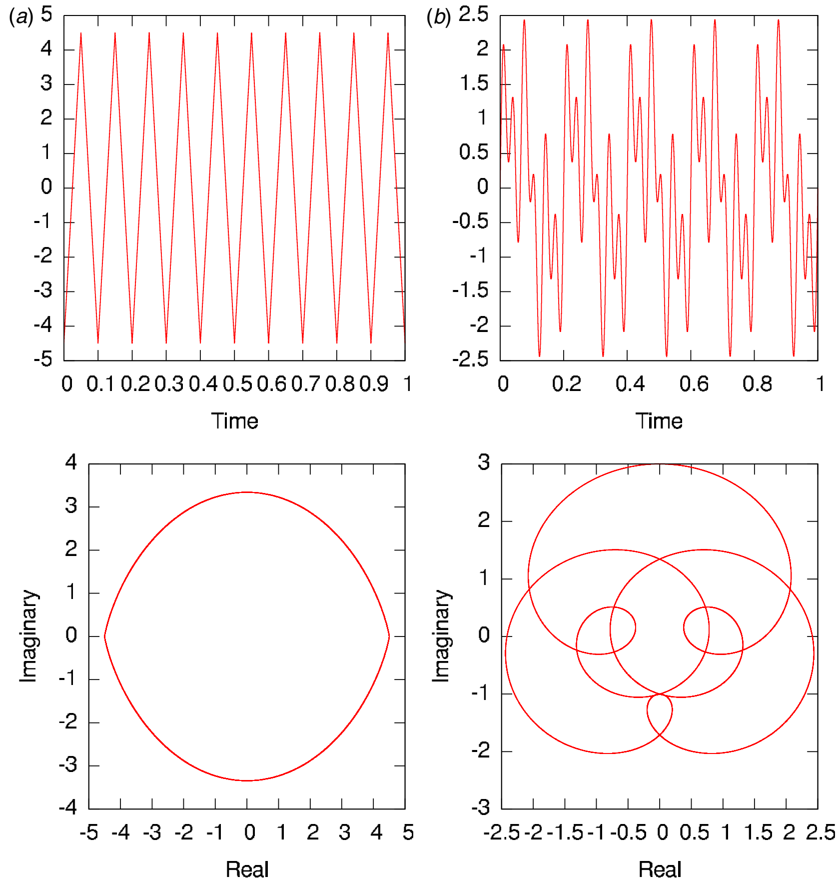


Figure 6. Sample analytic signals from real-valued signals. The first example is a perfectly periodic mono-component signal. The structure of the projected analytic signal is that of an orbit centred at the origin, with shape reflecting the shape of the oscillations. The phase of one point is the angle of that point in polar coordinates. Even though instantaneous phase can always be mathematically defined, it does not always carry a physical meaning. In the second example, there are different points within one single period with equal phase value due to the loops. Upper row: *(a)* $\sin(2\pi 5t)$, *(b)* $\sin(2\pi 5t) + \sin(2\pi 13t)$. Lower row: complex plane projections along the time axis of the corresponding analytic signals of the upper row.

From these, we calculate the phase difference between $m^{(k)}$ and $m^{(l)}$ making use of (11) as

$$\Phi^{(kl)} = \left\{\Phi_i^{(kl)} = \phi_i^{(k)} - \phi_i^{(l)} = \text{angle}\left(Z_i^{(k)} \cdot Z_i^{(l)*}\right)\right\}, \quad (13)$$

where the product with the complex conjugate is used to subtract the angle of $Z_i^{(l)}$ from the angle of $Z_i^{(k)}$. Figure 8(*c*) shows the phase differences between modules adjacent to module number 8. There is a clear transient period at the beginning of the simulation in which modules are not synchronized. During this time, the system explores its state space trying to find a stable state. After the transient, the phase differences between modules 8 and 7 and between 9 and 8 evolve in similar manners (see figure 9, where phase difference between consecutive modules is kept constant during all simulations). At the steady state, phase differences are maintained constant. Other modules show similar behaviours.

5.1.3. Frequency. Having defined the instantaneous phase of a signal in (11), we now define the instantaneous frequency of a signal as

$$\omega_i = \phi_i - \phi_{i-1}. \quad (14)$$

Figure 8(*d*) displays the frequency behaviour of module 8 of this CPG. The mean frequency is kept constant during the whole simulation except for an initial transient. Intra-cycle frequency, however, is not constant. That means that the speed at which the signal changes is not constant within one cycle.

5.2. Case study: an open topology

In the previous section we have addressed the analysis of a particular CPG built on bursting neurons, with strong inhibition and a non-open topology. The result is that all modules are capable of finding a stable oscillatory state, with constant amplitude and frequency, and with a constant phase difference between them.

We will repeat in this section the same analysis for an open topology. The basic structure is similar to that shown in figure 7, except that the border synapses have been removed. This way, border modules do not receive any input from other modules. Figure 10(*a*)–(*d*) show the results of the analysis applied to this open topology. Border modules oscillate at a frequency slightly higher than the rest of the modules because there is no inhibitory synapse acting on them. This difference in frequency prevents synchronization between modules. Figure 10(*c*) clearly shows how the phase difference between modules 2 and 1 (border module) drifts constantly.

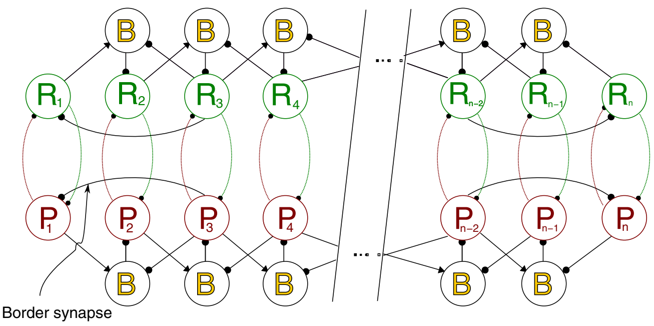


Figure 7. The 'bistable' CPG. The architecture for each module is conserved. Inter-module coordination mechanisms are shown in thicker trace. Arrow terminated lines are excitatory synapses; ball terminated lines are inhibitory synapses. This CPG is based on the assumption that to achieve a stable firing sequence, each neuron may only fire in an allowed time window, defined by the bursting activity of both its neighbours. 'B' neurons are bistable neurons. When they receive excitation they enter a tonic spiking regime. If they receive inhibition, they will enter a silent regime. Under absence of input, they will remain in the same state they were. 'B' neurons effectively restrict the allowed intervals through a strong synapse. When a 'B' neuron is in its active state, it will completely inhibit one 'P' (respectively 'R') neuron until it goes back to the silent regime. All parameters as in figure 5(*b*). Synapses from 'B' to 'P' and 'R': $E_{\mathrm{syn}} = 10$, $g_{\mathrm{syn}} = 20$.

Inner modules 2 and 3 cannot synchronize, yet their phase differences diverge in a lesser degree. The result is that the CPG fails to generate an efficient locomotion in the robot (see figure 11, where phase difference between consecutive modules suffers a slight shift during the simulation and is not kept constant).

6. Real hardware analysis

With the purpose of illustrating how certain biological principles can provide design guidelines for artificial CPG design, we have implemented a simulated CPG and analysed its stability. Now that we have concluded that the simulated CPG is stable, we target a real-world testbed robot with eight modules [31]. We refer the reader to section 4 for an extensive explanation both of the architecture of the robot and the CPG design process.

The simulation procedure is as follows: the CPG software is run offline during 1000 000 simulation steps. A program then reads each simulation step as a single line composed of the target positions for each motor. It selects one of every five lines so as to have an acceptable speed of locomotion, suitable for later video processing. Higher speeds can be achieved using a lower sampling rate, for instance one out of thirty simulation steps. Each target position is sent along with the number of the target motor over RS232, to a controller board (http://www.iearobotics.com/proyectos/skypic/skypic.html) running a program that generates a PWM signal that positions the motor at the specified angle.

We have carried out two tests to illustrate how the robot performs in the real world. The first one regards the stability of steady state locomotion of the real robot. In the second one, the robot is subject to a strong noisy perturbation, and then its recovery is analysed. Results are extracted from high-definition (hdv) video recordings (see figure 12) using a Sony HDR-HC9. Shooting was done from a distance of 2.2 m, and then videos were downloaded to a computer using dvgrab 3.5 on a linux machine. Uncompressed tiff frames were extracted from the video using ffmpeg, and then converted to uncompressed JPG using ImageMagick. A video-tracking software has been used on these frames to extract the positions of motor markers (figure 12).

6.1. *First test: free run of the CPG*

The first experiment consists of a free run simulation. We start the CPG with arbitrary initial conditions and let it evolve freely, with no perturbations. The output of the CPG is sent to the robot, which is recorded in video. Direct observation of the robot reveals that it performs an undulatory movement with steady forward locomotion. To illustrate this, we track the marker of one of the central modules

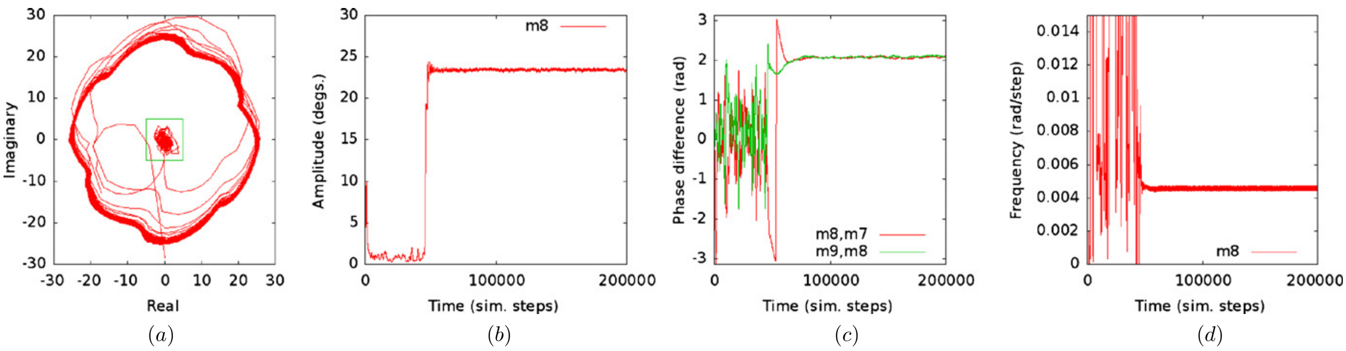


Figure 8. Analysis of the signal generated by the bistable CPG for module number 8. The system displays an initial transient period (marked with a green box in (*a*)) in which oscillations are not yet stable. After this, the system achieves a stable state in which amplitude, phase difference with neighbour modules and frequency remain constant. Signals generated for other modules show similar behaviour, with possibly different transient periods but similar steady states. (*a*) Projection of the analytic signal generated by the CPG for module number 8. The shape is similar to the mono-component signals in figure 6. This gives an idea of the oscillatory behaviour of the signal. The points in the centre correspond to an initial transient period before the system achieves a stable state. (*b*) Amplitude envelope of the signal. (*c*) Phase differences between modules 8 and 7 and between modules 9 and 8. (*d*) Frequency of the signal.

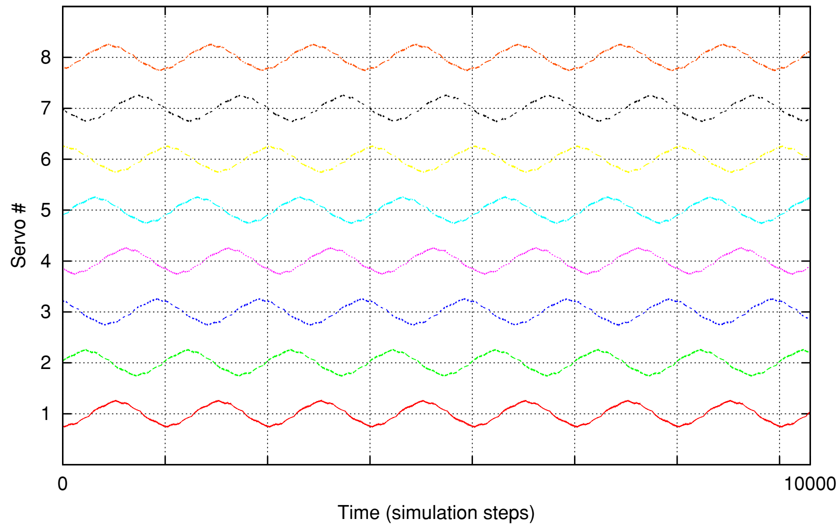
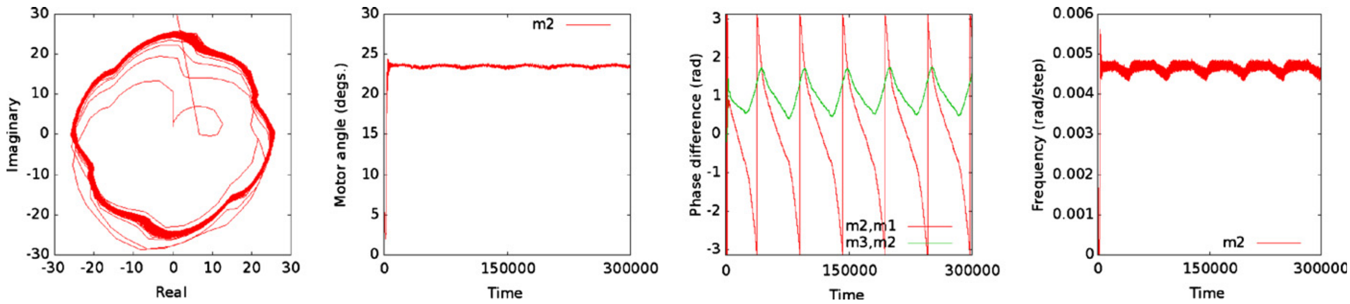


Figure 9. Snapshot of the CPG analysed in figure 8. After an initial transient interval, the CPG runs synchronized and is working in the stable regime. All signals show a steady oscillatory behaviour with constant amplitude and frequency. Grid marks are set at intervals of 1370 time steps, approximately one period of oscillation. All signals maintain a constant phase relationship among them and elicit a stable locomotion.



(*a*) (*b*) (*c*) (*d*)

Figure 10. Analysis of the signal generated for module number 2 by a CPG with an open topology. In this simulation the transient period is very short and the system quickly finds an oscillatory state. However, the system fails to maintain a stable rhythm and instead shows a metastable sequence. There are periodic variations in frequency that, in turn, prevent adjacent modules from keeping a constant phase difference. (*a*) Projection of the analytic signal generated by the CPG for module number 2. (*b*) Amplitude envelope of the signal. (*c*) Phase differences between modules 2 and 1 and between modules 3 and 2. There is a clear constant phase difference shift which prevents the necessary phase locking for efficient locomotion. (*d*) Frequency of the signal.

and analyse the trace offline (see figure 13). In order to gain more insight about the performance of the robot, the signal is decomposed into its two coordinates with respect to time (figure 14). The vertical component of the module is periodic: this reveals that the movement of the marker is indeed oscillatory; due to the geometry of the robot the trajectory is not constant, with some plateaus of resting activity, but the overall behaviour is sinusoidal. Horizontal forward velocity is almost constant, with a mean value of approximately 1.4 cm s^{-1}, again with small plateaus of quietude due to the robot's geometry.

This experiment confirms that our CPG can effectively drive the real robot and perform a steadily undulatory locomotion, with uniform forward velocity.

For comparison purposes we include figure 15 in which the analysis outlined in the previous section is performed on the vertical displacement of the marker shown in figure 13. This signal does not meet the requirements for the instantaneous phase to have a valid physical interpretation for every time instant (indeed, there is an artefact, surrounded by a rectangle, in figure 15 whose interpretation would be that the phase went backwards in time, which is physically impossible). However it is still valid to infer the mean behaviour of the phase difference between adjacent modules with respect to time. Subfloat 15(*a*) shows the corresponding analytical signal, obtained using the Hilbert transform, projected on the complex plane. It is seen that the trace of the marker is oscillatory, stable and periodic. What is more, the module at hand is seen to keep a reasonably constant phase difference with its neighbours, as seen in panel 15(*b*). Small oscillations in this panel are due to the fact that instantaneous frequency is not constant within one oscillation cycle. Effectively, coordination is guaranteed by the CPG, which can easily handle these small differences.

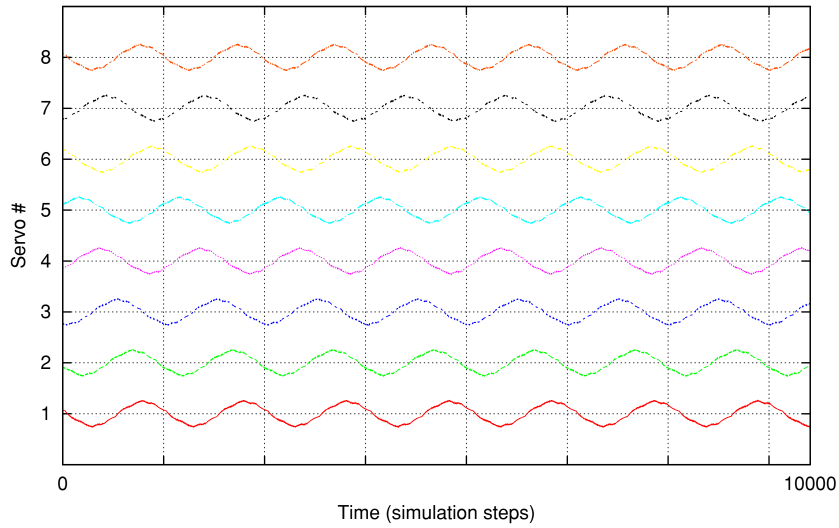


Figure 11. Snapshot of the CPG analysed in figure 10. This CPG is not built with an open topology. That is, there are two modules in the topology that do not receive any feedback from the rest of the circuit. For this reason, the circuit is not able to maintain stable synchronization between signals, and their frequencies diverge periodically in time as shown in figure 10(*c*). For this time slot, the three last signals show a markedly skewed phase in relationship to the other servos. Grid marks set at intervals of 1350 time steps, approximately the mean period of all signals. However, while signal number 2 has a period that fits the mean, signal number 8 is clearly oscillating faster.

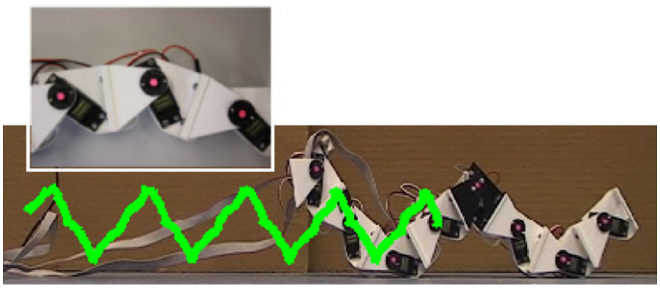


Figure 12. High-definition motion tracking. Video was recorded using a Sony HDR-HC9 in high definition (hdv), shooting from a distance of 2.2 m, and downloaded to a computer using dvgrab 3.5 on a Linux machine. Uncompressed tiff frames where extracted from the video using ffmpeg, and then converted to uncompressed JPG using ImageMagick. The green trace corresponds to one of the middle segments of the robot. In the upper-left corner the inset shows a closeup of the modules with the position markers used for locomotion tracking. Original footage is available as supplementary data, available at stacks.iop.org/BB/6/016006/mmedia.

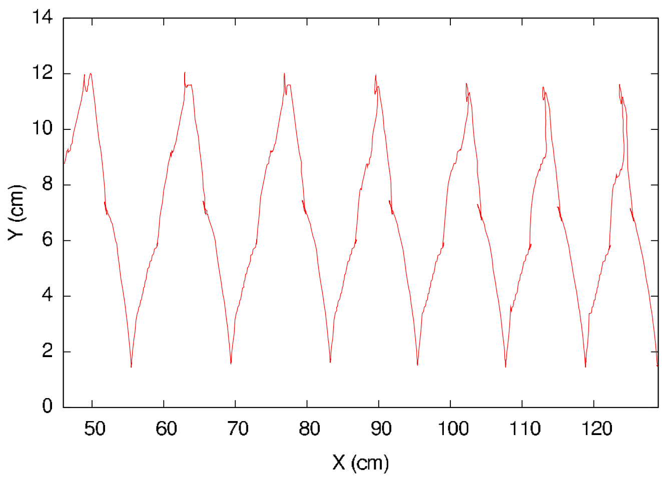


Figure 13. Video tracking of the middle segment of the real robot, locomoting in the stable regime. The *X* coordinate represents the horizontal displacement from the left border of the recording area; the *Y* coordinate represents vertical displacement from the ground. The trace of the point is clearly periodic, a consequence of the periodicity of the controller CPG.

6.2. Second test: recovery after perturbation

In the second experiment, simulation begins with the CPG at arbitrary initial conditions, as in the first one. The CPG is left to evolve and settle at a stable steady state. At a given interval in time, a noisy stimulus (random variable from a uniform distribution in the interval [0, 30)) is applied to all neurons in the CPG controlling the real robot. Analysis is performed from the onset of the perturbation to a point where locomotion is again stable (figure 16 and 17). During perturbation the robot lies flat on the ground without making any movement at all, not even small trembling of the motors. Once the perturbation is over, neurons resume their activity and a new synchronization process begins. During this process locomotion is ineffective and the robot undulates in place, without travelling (see the rectangle in figure 16). After a short time synchronization is achieved and an effective locomotion is re-established.

7. Discussion

CPGs are neural networks responsible for rhythmic behaviour in animals. Based both on intrinsic neuron dynamics and connectivity properties, CPGs generate and coordinate rhythmic movements in a robust yet flexible manner modulated by sensory feedback. Artificial CPG circuits are particularly suitable for the design of autonomous modular robots, as CPG

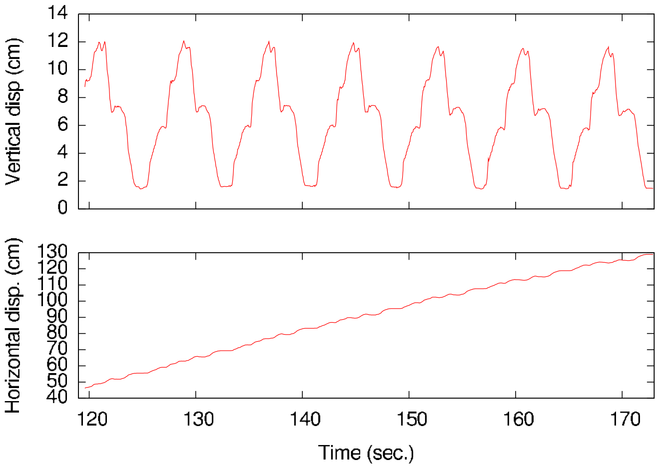


Figure 14. Video tracking of the real robot: vertical and horizontal coordinates with respect to time. The bottom panel shows a steady forward locomotion with an approximate overall speed of 1.4 cm s^{-1}.

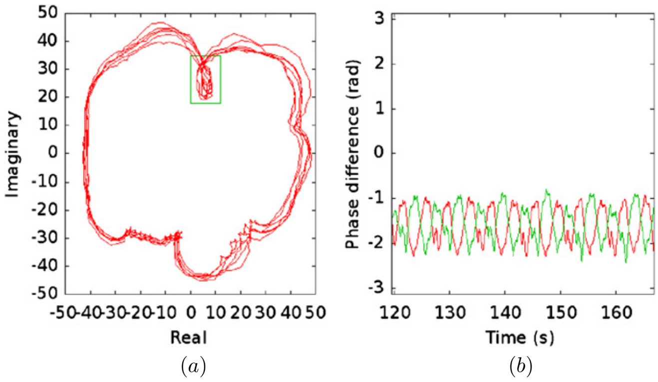


Figure 15. (*a*) Hilbert transform of the vertical displacement of the central module of the real robot (panel (*a*) in figure 14). The analysed signal presents some anomalies (green rectangle) that make instantaneous frequency interpretation invalid (here the phase would go backwards, which is not a valid physical interpretation); however, the analysis is still useful to convey an overall idea of the behaviour of the system. The movement of the central module is rhythmically stable, with constant amplitude and frequency. (*b*) Phase difference of the central module with its predecessor and successor modules. Since the analysis yields nonconstant instantaneous frequency, instantaneous phase difference oscillates in every cycle of the rhythm. However, the mean phase difference is clearly bounded (mean: $1.55 \approx \pi/2$ rad; std: 0.39 rad).

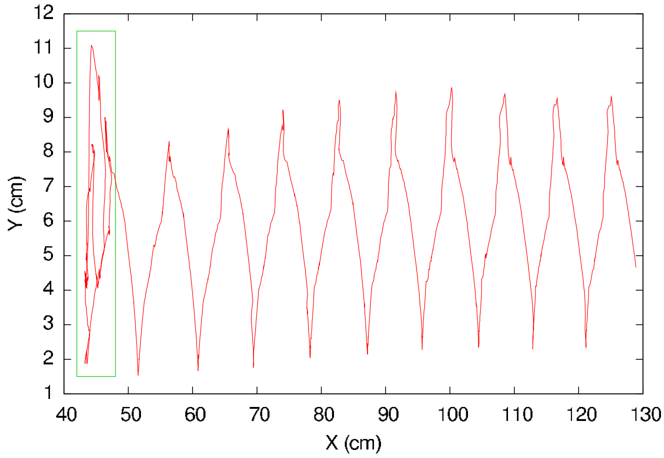


Figure 16. Recovery after a disruptive noise is applied to every neuron controlling the real robot. A normal simulation is carried out. At a given time a high level of noise (random uniform distribution in the interval [0, 30)) is injected into every neuron of the CPG. As a result of the noise, neurons stop displaying their normal bursting activity and the CPG generates no oscillations at all. Right after the noisy stimulus is released, neurons go back to their bursting behaviour. The robot finds itself in an uncoordinated state (activity surrounded by a rectangle). Eventually, the robot resynchronizes itself and resumes forward locomotion.

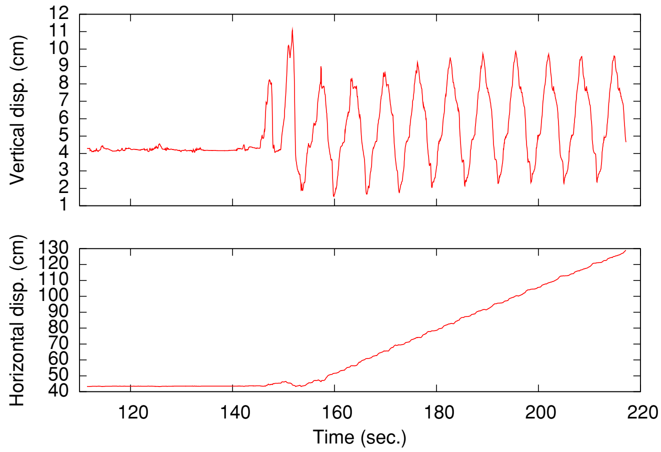


Figure 17. Recovery after a disruptive noise is applied to every neuron controlling the real robot. A noisy stimulus is applied that disrupts the activity of all neurons in the CPG. During this stimulus the robot lies still on the ground without making any move. After noise is removed, the robot begins searching for a synchronized state. After only two oscillations, the robot is again moving forward at an approximate speed of 1.5 cm s^{-1}.

control fully fits the idea of having variable number of modules that are organized by the same scalable principles.

In this paper, we have analysed different strategies found in recent research on living CPGs and tested them on an example robotic platform. Then, we have dissected the problem of designing a bio-inspired CPG for a modular robot into individual independent problems. We have formulated each problem in a way that is very general and should be easily adapted for other robotic platforms. To begin with, we have exposed the need for a mechanism to robustly encode locomotion commands and still be flexible enough to negotiate an effective rhythm, and have provided a solution based on neuron models with multiple time scales. We have provided a mechanism based on non-open topologies with mutual inhibition that generates adaptable oscillations to control one individual module.

The interface between neural *language* and servo actuation is a bio-inspired integrator motoneuron model. It is responsible for the decoding of the dynamics of the CPG, and translating it into actual servo position commands. Communication between servo and CPG to promote adaptability is also possible. We have shown in a simulation that the oscillator can entrain a servo in a wide range of situations, ranging from very fast servos to very slow ones. In the appendix, we illustrate an implementation of feedback to obtain entrainment. There, the oscillator shows its ability to adapt its period well over several orders of magnitude, depending on the capacity of the servo to follow it. This ability is due to the fact that the neurons that compose the oscillator have multiple time scales: the slow subsystem may prolong its activity while the fast subsystem keeps on oscillating. This translates into longer bursts, while individual spikes are emitted at a constant frequency.

Then, we have built a modular and scalable inter-module architecture, a non-open topology based on inhibitory connections to first neighbours. This topology, together with the intrinsic properties of the neurons, gives rise to a *winnerless competition* [8] dynamics that allows modules to exchange information and autonomously organize. Non-open topologies, widespread in living CPGs, have proved to be the most effective way of achieving rhythm autoregulation [6, 7]. We have quantitatively assessed the stability and robustness of the rhythms produced by the proposed non-open architecture. In particular, we have confirmed that under normal conditions, the CPG is able to maintain stable amplitudes, frequencies and phase differences. Moreover, it is able to recover from perturbations and quickly regain normal activity.

Finally, hardware testing has confirmed that a CPG built with these bio-inspired strategies can effectively drive a real robot. After a short time of negotiation, the robot undulates in forward locomotion with a steady speed. After a strong perturbation to the CPG (running in open loop), the robot recovers and resumes forward locomotion unaltered.

Recent theoretical and experimental evidence have shown that the interplay between CPG network properties and intrinsic activity in single neurons provides robust and reproducible transient dynamics which is key to an autonomous coordinated response [5, 8]. Our strategy to build autonomous robots is based on connecting bio-inspired building blocks with an appropriate topology, and we rely on the properties of the elements so connected to negotiate an effective locomotion plan.

The emphasis has been on a qualitative approach, where the focus has been the selection of appropriate dynamical properties of the building blocks of our CPG. We know that living CPGs achieve similar behaviours with different combinations of parameters [39, 40]. So despite the apparent complexity of this approach, the fact that more parameters are introduced in the design results in higher flexibility, rather than increased complexity.

Living CPGs receive feedback from other CPGs, higher centres and sensory receptors. Autoregulatory mechanisms are essential in the ability of a CPG to adapt to external circumstances. The proposed bio-inspired strategies are also adequate for closed loop interaction and can lead to oscillation period autoregulation, as we illustrate in the appendix.

Advances in CPG research in recent years provide new bio-inspiration for robotic design. We believe that the proposed design strategies can lead to CPG control paradigms for autonomous locomotion that are less architecture specific and provide solutions that present wider working regions in the parameter space of the models. We plan to further investigate and take advantage of how biological systems are able to maintain their invariants and incorporate these new ideas into autonomous robotic control.

Acknowledgments

We thank Juan González Gómez for his inspiration, his dedicated work and, of course, for his robots. This work was supported by MICINN BFU2009-08473, TIN 2007-65989, TIN-2010-19607 and CAM S-SEM-0255-2006. FH-C acknowledges support from an FPU-UAM grant.

Appendix. Implementing entrainment between a servo and its controller

The true power of a CPG controller lies in its ability to maintain a certain rhythmic activity while adapting itself to external conditions. The particular type of adaptation will depend upon the configuration of the CPG. In this section, we will briefly introduce a case study for a particular configuration. We have performed several experiments to show how a modular oscillator would react when coupled to different servos in different working conditions, without modifying the configuration of the CPG itself.

We simulate a servo using the following equation:

$$e_n = u_n - s_n \tag{A.1}$$

$$s_{n+1} = s_n + \mu e_n, \tag{A.2}$$

where s_n is the servo position at time step n; u_n is the control parameter indicating the target position and μ is a velocity constant ranging from 0 to 1, whereby values of μ closer to 0 mean a very slow servo (up to infinitely slow, i.e. motionless if $\mu = 0$) and values closer to 1 mean a very quick servo (up to infinitely quick, with only one time step of delay if $\mu = 1$).

Coupling between the simulated servo and the CPG is performed by using $u_n = m(t)$ from (6) as the control parameter. Then, for the promotor and remotor neurons of the oscillator, the equation of synaptic input would read (compare to the original in (3))

$$I_n^{\mathrm{p}} = g \cdot r_n^{\mathrm{p}} \cdot \left(x_n^{\mathrm{p}} - E_{\mathrm{syn}}\right) + Ae_n \tag{A.3}$$

$$I_n^{\mathrm{r}} = g \cdot r_n^{\mathrm{r}} \cdot \left(x_n^{\mathrm{r}} - E_{\mathrm{syn}}\right) - Ae_n, \tag{A.4}$$

where ‘p’ and ‘r’ denote whether the receiving neuron is the promotor or the remotor neuron, respectively, and A is a scaling factor that represents the importance of feedback.

Effectively, a modular CPG can be entrained with a servo in this manner. Figure A1 depicts three working cases in which three different servos are simulated, each one with different working velocity. In all three cases the simulated CPG is working with the same parameters. In the first case, a very slow servo is simulated. The CPG sustains its activity for as long as the servo needs to reach the final position. As faster servos are simulated, the CPG oscillates ostensibly faster, up to its nominal frequency for servos with μ close to 1. Figure A2

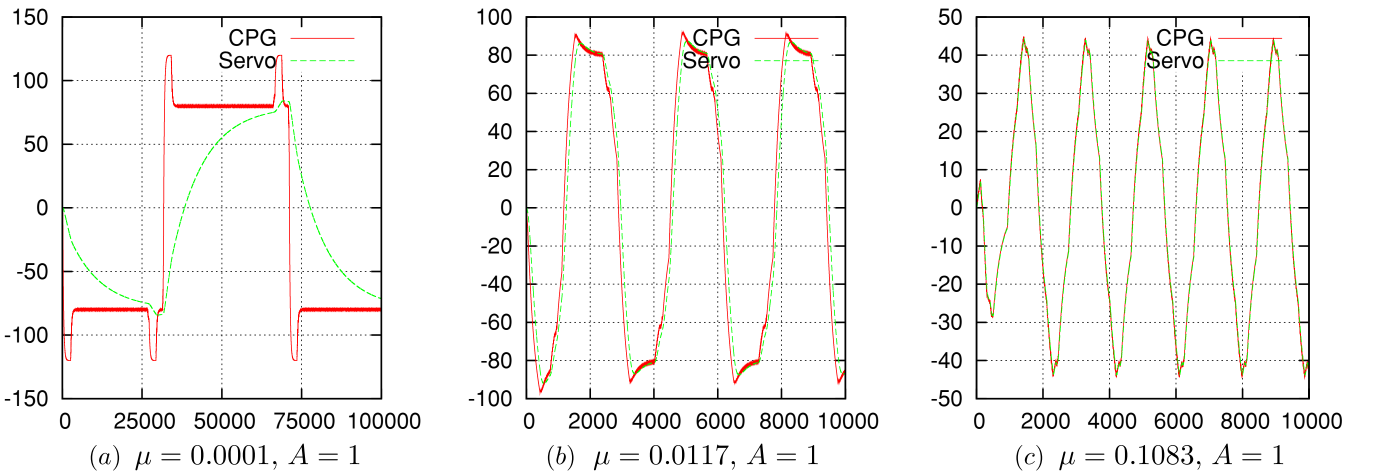


Figure A1. Entrainment between a simulated servo and a modular CPG. Panels display CPG activity and corresponding servo position (in degrees). The CPG reshapes its activity to accommodate the activity of the servo. Values of μ closer to zero indicate a slower servo; values closer to 1 indicate a faster servo. The frequency of the CPG may be slowed down up to some orders of magnitude. Note the change in the range of the axes. (*a*) An extreme example in which both the remotor and the promotor neurons *wait* for the servo to reach the targeted position. (*b*) The CPG runs with a period of approximately 1500 time steps without feedback; in this example, the CPG has clearly adapted its activity to a period well over 2000 time steps. (*c*) With a fast enough servo, the CPG runs close to its nominal regime.

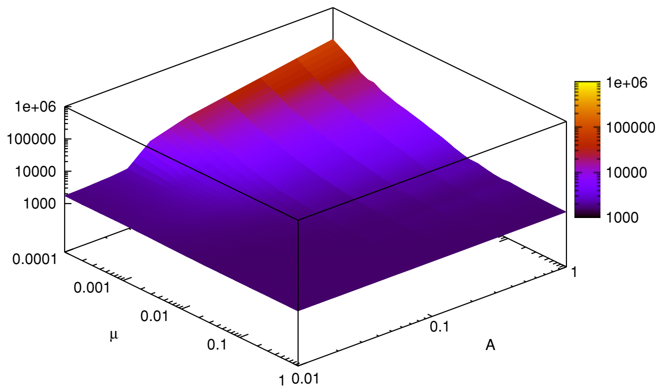


Figure A2. Period of oscillation (in time steps) of a servo in an entrained setup, depending on servo capability and feedback strength (note logarithmic scales in all axes). μ indicates the responsiveness of the servo, ranging from 0 (motionless) to 1 (au), an infinitely fast servo; A is a scaling factor for feedback, ranging from 0, no feedback at all, to 1 (au); the error is fed back to the CPG. Clearly, if the servo is fast enough (in the region $\mu \in [0.1, 1]$) the period of oscillation does not depend on feedback strength. For slower servos, feedback strength is definitely important in how long the CPG will *wait* for it to reach the target position: for lower values of A, the CPG is oscillating at its nominal frequency due to insufficiently strong feedback, while for higher values of A, the position error of the servo is taken into account, as is the case in figure A1(*a*).

shows the simulated parameter region and how the CPG is able to adapt its working period several orders of magnitude beyond its nominal period.

The fact that the neurons that build the oscillator have multiple time scales is key to the adaptability of the CPG. With the proposed bio-inspired design, the CPG uses external feedback to self-regulate its period, extending the length of neural bursts. See for instance figure A1(*a*), where the CPG is held in a steady state around 90°, *waiting* for the servo to reach that position. In this case, the fast subsystem (x_n in (1*b*)) oscillates unperturbed. If feedback is high enough, the slow subsystem (y_n in (1*c*)) is kept at a steady equilibrium that prevents the burst from terminating.

References

[1] Grillner S 2006 Biological pattern generation: the cellular and computational logic of networks in motion *Neuron* **52** 751–66
[2] Marder E and Bucher D 2007 Understanding circuit dynamics using the stomatogastric nervous system of lobsters and crabs *Annu. Rev. Physiol.* **69** 291–316
[3] Smarandache C, Hall W M and Mulloney B 2009 Coordination of rhythmic motor activity by gradients of synaptic strength in a neural circuit that couples modular neural oscillators *J. Neurosci.* **29** 9351–60
[4] Marder E and Bucher D 2001 Central pattern generators and the control of rhythmic movements *Curr. Biol.* **11** R986–96
[5] Selverston A I., Rabinovich M I, Abarbanel H D I, Elson R, Szucs A, Pinto R D, Huerta R and Varona P 2000 Reliable circuits from irregular neurons: a dynamical approach to understanding central pattern generators *J. Physiol. Paris* **94** 357–74
[6] Huerta R, Varona P, Rabinovich M I and Abarbanel H D 2001 Topology selection by chaotic neurons of a pyloric central pattern generator *Biol. Cybern.* **84** L1–8
[7] Stiesberg G R, Reyes M B, Varona P, Pinto R D and Huerta R 2007 Connection topology selection in central pattern generators by maximizing the gain of information *Neural Comput.* **19** 974–93
[8] Rabinovich M I, Varona P, Selverston A I and Abarbanel H D I 2006 Dynamical principles in neuroscience. *Rev. Mod. Phys.* **78** 1213–65
[9] Reyes M, Huerta R, Rabinovich M and Selverston A 2008 Artificial synaptic modification reveals a dynamical invariant in the pyloric CPG *Eur. J. Appl. Physiol.* **102** 667–75
[10] Szucs A, Pinto R D, Rabinovich M I, Abarbanel H D and Selverston A I 2003 Synaptic modulation of the interspike interval signatures of bursting pyloric neurons *J. Neurophysiol.* **89** 1363–77

[11] Latorre R, Rodríguez F B and Varona P 2006 Neural signatures: multiple coding in spiking-bursting cells *Biol. Cybern.* **95** 169–83

[12] Kurokawa H, Tomita K, Kamimura A, Kokaji S, Hasuo T and Murata S 2008 Distributed self-reconfiguration of M-TRAN III modular robotic system *Int. J. Robot. Res.* **27** 373–86

[13] Yim M, Duff D G and Roufas K D 2000 PolyBot: a modular reconfigurable robot *Proc. IEEE Int. Conf. Robotics and Automation* vol 1 pp 514–20

[14] Ijspeert A 2008 Central pattern generators for locomotion control in animals and robots: a review *Neural Netw.* **21** 642–53

[15] Degallier S and Ijspeert A J 2010 Modeling discrete and rhythmic movements through motor primitives: a review *Biol. Cybern.* **103** 319–38

[16] Ayers J and Witting J 2007 Biomimetic approaches to the control of underwater walking machines *Phil. Trans. R. Soc.* A **365** 273–95

[17] Arena P, Fortuna L, Frasca M and Patane L 2005 A CNN-based chip for robot locomotion control *IEEE Trans. Circuits Syst.* **52** 1862–71

[18] Chung S-J and Dorothy M 2009 Neurobiologically inspired control of engineered flapping flight *J. Guid. Control Dyn.* **33** 440–53

[19] Seo K, Chung S-J and Slotine J J 2010 CPG-based control of a turtle-like underwater vehicle *Auton. Robots* **28** 247–69

[20] Manoonpong P, Geng T, Kulvicius T, Porr B and Wörgötter F 2007 Adaptive, fast walking in a biped robot under neuronal control and learning *PLoS Comput. Biol.* **3** 1305–20

[21] Aoi S and Tsuchiya K 2006 Stability analysis of a simple walking model driven by an oscillator with a phase reset using sensory feedback *IEEE Trans. Robot.* **22** 391–7

[22] Ijspeert A J and Kodjabachian J 1999 Evolution and development of a central pattern generator for the swimming of a lamprey *Artif. Life* **5** 247–69

[23] Kamimura A, Kurokawa H, Toshida E, Tomita K, Murata S and Kokaji S 2003 Automatic locomotion pattern generation for modular robots *IEEE Int. Conf. Robotics and Automation* **vol 1** pp 714–20

[24] Crespi A and Ijspeert A J 2008 Online optimization of swimming and crawling in an amphibious snake robot *IEEE Trans. Robot.* **24** 75–87

[25] Sproewitz A, Moeckel R, Maye J and Ijspeert A J 2008 Learning to move in modular robots using central pattern generators and online optimization *Int. J. Robot. Res.* **27** 423–43

[26] Rulkov N F 2002 Modeling of spiking-bursting neural behavior using two-dimensional map *Phys. Rev.* E **65** 041922

[27] Shilnikov A L and Rulkov N F 2003 Origin of chaos in a two-dimensional map modeling spike-bursting neural activity *Int. J. Bifurcation Chaos* **13** 3325–40

[28] Stevens W P, Myers G J and Constantine L L 1974 Structured design *IBM Syst. J.* **13** 115–39

[29] Orlovsky G, Deliagina T G and Grillner S 1999 *Neuronal Control of Locomotion* (Oxford: Oxford University Press)

[30] Destexhe A, Mainen Z F and Sejnowski T J 1994 An efficient method for computing synaptic conductances based on a kinetic model of receptor binding *Neural Comput.* **6** 14–8

[31] González-Gómez J, Zhang H, Boemo E and Zhang J 2006 Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules *CLAWAR 06: 9th Int. Conf. Climbing and Walking Robots (Brussels, Belgium, 12–14 September 2006)*

[32] Herrero-Carrón F, Rodríguez F B and Varona P 2010 Studying robustness against noise in oscillators for robot control *Int. Conf. Automation, Robotics and Control Systems* pp 58–63

[33] Thuma J B, Morris L G, Weaver A L and Hooper S L 2003 Lobster (*Panulirus interruptus*) pyloric muscles express the motor patterns of three neural networks, only one of which innervates the muscles *J. Neurosci.* **23** 8911–20

[34] Zhurov Y and Brezina V 2006 Variability of motor neuron spike timing maintains and shapes contractions of the accessory radula closer muscle of *Aplysia J. Neurosci.* **26** 7056–70

[35] Brezina V, Orekhova I V and Weiss K R 2000 The neuromuscular transform: the dynamic, nonlinear link between motor neuron firing patterns and muscle contraction in rhythmic behaviors *J Neurophysiol.* **83** 207–31

[36] Herrero-Carrón F, Rodríguez F B and Varona P 2010 Dynamical invariants for CPG control in autonomous robots *7th Int. Conf. Informatics in Control, Automation and Robotics* vol 2 ed J Filipe, J A Cetto and J-L Ferrier pp 441–5

[37] Oppenheim A V, Schafer R W and Buck J R 1999 *Discrete-Time Signal Processing* 2nd edn *(Prentice-Hall Signal Processing Series)* (Englewood Cliffs, NJ: Prentice-Hall) chapter 11, pp 775–802

[38] Huang N E, Shen Z, Long S R, Wu M C, Shih H H, Zheng Q, Yen N C, Tung C C and Liu H H 1998 The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis *Proc. R. Soc.* A **454** 903–95

[39] Prinz A A, Bucher D and Marder E 2004 Similar network activity from disparate circuit parameters *Nat. Neurosci.* **7** 1345–52

[40] Marder E and Goaillard J-M 2006 Variability, compensation and homeostasis in neuron and network function *Nat. Rev. Neurosci.* **7** 563–74