Rethinking Central Pattern Generators: A General Approach [1]

Chris Eliasmith [a] and Charles H. Anderson [b]

[a] *Department of Philosophy, Philosophy-Neuroscience-Psychology Program, Washington University in St. Louis, Campus Box 1073, One Brookings Drive, St. Louis, MO 63130, chris@twinearth.wustl.edu*

[b] *Department of Anatomy and Neurobiology, Washington University School of Medicine, St. Louis, MO 63110, cha@shifter.wustl.edu*

Abstract

Central pattern generators (CPGs) have traditionally been modeled as sets of coupled bistable oscillators [2]. We present a framework for constructing models which avoid the shortcomings of these traditional models, while remaining biologically plausible. We demonstrate our approach by generating a novel model of lamprey locomotion. However, we suggest that the methods presented here can be more generally applied to modeling any neural system which produces and controls dynamic patterns of forces.

Key words: Central pattern generator; CPG; Neural computation; Lamprey locomotion; Coupled oscillators

1 Introduction

Repetitive behavioral patterns such as swimming, flying, chewing, breathing, scratching and walking have long been a mainstay of motor research in neuroscience. A pivotal concept in the analysis of such rhythmic behavior is that of the central pattern generator (CPG) [3]. A CPG is a group of neurons that can produce rhythmic patterns without sensory input. It has been observed that reciprocally connected networks with inhibitory weights (i.e. bistable oscillators) are able to produce CPG-like rhythmic patterns. Traditionally, it has

[1] This research has been supported by the McDonnell Center for Higher Brain Function at Washington University (both authors) and the Social Sciences and Humanities Research Council of Canada (Eliasmith).

been assumed that because such networks produce a behavior similar to that seen in CPGs, they are good models of CPGs. However, there are a number of difficulties with such phenomenological models. For example, Marder et al. have shown that these networks in no way guarantee oscillations and often produce synchrony: a deadly result for an organism [1]. As well, Wannier et al. have noted that it is difficult to control the direction and frequency of the oscillations arising from coupled networks of bistable oscillators - a common model of leech and lamprey locomotion [6].

In the remainder of this paper, we present an alternative approach to modeling CPGs. In the next section we describe a five step process which can be used as a general methodology for constructing models of motor control systems. We then apply this process to understanding lamprey locomotion. We argue that given the strengths and successes of the resulting simulation, this methodology might prove generally useful as a means of modeling neural systems which generate dynamic force patterns.

2 A Five Step Process Used to Model Lamprey Swimming

The iterative five step process we apply in the remainder of this paper is as follows:

(1) Determine the forces to be generated by the neural system; $F(x,t)$
(2) Define the representation to be used (via encoding and decoding rules); $F(x,t) = \sum_n A_n(t)\Phi_n(z)$
(3) Create a set of dynamical equations that lead to the desired $A_n(t)$,

$$\frac{d\mathbf{A}}{dt} = \mathcal{G}\left(\mathbf{A}(t), \mathbf{U}(t), t\right) \tag{1}$$

(4) Implement and test the model; and
(5) Redefine the representation and repeat 2-4 until the system is defined in the space of neuronal activities.

As we use this method to construct a model of lamprey locomotion, we indicate which step we are currently engaged in with a label in bold (e.g. **Step 1**).

Step 1 A simplified model of lamprey swimming leads to a model of the tension given as

$$T(z,t) = \kappa[\cos(\omega t - 2\pi z/L) - \cos(\omega t))], \tag{2}$$

where $v = \omega * L/2\pi$ defines the velocity, L is the length of the lamprey, and κ is a scaling parameter. This equation specifies the tensions which must be generated by the nervous system via the muscles. When the equation is satisfied, the lamprey will swim in an anguilliform mode (i.e. with a traveling wave).

Step 2 We now define a representation of this pattern of forces in terms of temporal coefficients, $A_n(t)$ and spatial harmonic functions $\Phi_n(z)$.

$$T(z,t) = \kappa \sum_{n=0}^{N} A_{2n}(t)\cos(2\pi nz) + A_{2n+1}(t)\sin(2\pi nz), \tag{3}$$

where $A_0(t) = -\cos(\omega t)$, $A_1(t) = -\sin(\omega t)$, $A_2(t) = \cos(\omega t)$, $A_{\{n>2\}}(t) = 0$.

Step 3 Using this representation we create a set of dynamical equations that result in the desired $A_n(t)$.

$$\frac{d\mathbf{A}}{dt} = \mathbf{M}\cdot\mathbf{A}(t) = [\omega\mathbf{M}_\omega + \mathbf{M}_d + U(t)\mathbf{M}_s]\cdot\mathbf{A}(t). \tag{4}$$

where

$$\mathbf{M}_\omega = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{M}_d = \begin{bmatrix} -\alpha_0 & 0 & -\alpha_0 & 0 \\ 0 & 0 & 0 & 0 \\ -\alpha_0 & 0 & -\alpha_0 & 0 \\ 0 & 0 & 0 & -\alpha \end{bmatrix}, \text{ and } \mathbf{M}_s = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

This equation is a simpler form of the general control dynamics equation (1). We have broken down the transformation matrix $\mathbf{M}$ to give us precise control over the dynamics of the model. In particular, $\mathbf{M}_\omega$ controls the steady state oscillatory swimming dynamics; $\mathbf{M}_d$ uses two rate constants, α_0 to force the first and third Fourier amplitudes to be equal in magnitude and opposite in sign, and α to damp out high spatial frequencies; and $\mathbf{M}_s$ controls the start up behavior of the lamprey by inducing exponential growth in the desired amplitudes.

Step 4 Simulating this model results in the expected behavior. The lamprey swims in steady state with a traveling wave whose temporal frequency is controlled by ω. As well, high frequency harmonics are damped out and the wave's amplitude increases exponentially to some desired value at startup as controlled by $\mathbf{U}(t)$.

Step 5 Now that we have a simple working model, we 2-4, by projecting the current representation to a more neurologically reasonable one.

Step 5.2 To begin, we know that the control of muscles by the neural system is local along the length of the lamprey. This means we have a spatially segregated representation of the tension. To capture this aspect of lamprey anatomy, we define a spatially segregated representation. We use Gaussian functions to represent the tension over local regions ($\approx \sigma_g$) along the length of the lamprey.

$$T(z,t) = \kappa \sum_m a_m(t)\phi_m(z) = \kappa \sum_m a_m(t) \exp(-(z - m * dz)^2/\sigma_g^2), \quad (5)$$

where $a_m(t)$ is the amplitude of the mth Gaussian centered at the point $z_m = m * dz$.

We can now construct the projection operator Γ by projecting our previously constructed Fourier basis, $\Phi_n(z)$, onto our new Gaussian basis, $\phi_m(z)$ (i.e. $\Gamma_{nm} = \int \Phi_n(z)\phi_m(z)dz$). This projection operator allows us to "move between" the two representational spaces we have constructed.

Step 5.3 Next, we take advantage of Γ to transform the dynamical equations for the Fourier amplitudes $\mathbf{A}(t)$ into dynamical equations in the space of the Gaussian amplitudes $\mathbf{a}(t)$. We can write:

$$\frac{d\mathbf{a}(t)}{dt} = \mathbf{m} \cdot \mathbf{a}(t), \text{ where } \mathbf{m} = \Gamma^{-1}\mathbf{M}\Gamma. \quad (6)$$

Because of the presumed redundancy in the Gaussian representation, Γ^{-1} is the pseudoinverse of Γ.

Step 5.4 We can now simulate the lamprey's swimming in the Gaussian space using (6). Not surprisingly, the lamprey swims just as it did before. However, the coupling weights are more global than one might expect given the length of projections in the lamprey [2]. It is possible, however, to define the transformation M_ω using the gradient of the tension in z, which makes the coupling more local in the Gaussian space. This allows us to match the known neural projection data from the lamprey.

Step 5.5.2 We can now perform a second iteration of steps 2-4 by introducing a neuronal representation of the Gaussian amplitudes, $\mathbf{a}_m(t)$:

$$c_{mk} = \mathcal{F}_{mk}[a_m(t)] \quad \text{(encode)} \quad (7)$$

$$a_m(t) = \sum_k \alpha_{mk} c_{mk}(t) \quad \text{(decode)} \; . \quad (8)$$

Here, $\mathcal{F}_{mk}[]$ is the nonlinear spike generation process that *encodes* $a_m(t)$ into the firing rate of the neuron indexed by m, k. The α_{mk} *decode* the firing rate back to a Gaussian amplitude [4][5]. In this paper, we define the encoding process as $\mathcal{F}_{mk}[a_m(t)] = [g_{mk}a_m(t) + b_{mk}]_+$, where $[]_+$ stands for rectification. The response of the neuronal population is defined by assuming a plausible set of encoding parameters, g_{mk} (gain) and b_{mk} (bias), and then finding the optimal weights α_{mk} using a procedure such as singular value decomposition to minimize the square error of the decoding. Notably, this kind of analysis also works well for more complex forms of encoding, like that found in full conductance models of spiking neurons.

Step 5.5.3 The dynamics of the neuron firing rates, c_{mk}, can be found by first noting that $\tau_c \frac{dc_{mk}(t)}{dt} \approx -(c_{mk}(t) - c_{mk}(t+\tau_c))$ and then using the encoding and decoding relationships (7) and (8) and the Gaussian dynamics (6) as follows:

$$
\begin{aligned}
c_{mk}(t+\tau_c) = \mathcal{F}[a_m(t+\tau_c)] &\approx \left[a_m(t) + \tau_c \frac{da_m(t)}{dt}\right]_+ = [a_m(t) + \tau_c \mathbf{m} \cdot \mathbf{a}(t)]_+ \\
&\approx \left[g_{mk}(\sum_l \alpha_{ml} c_{ml}(t) + \tau_c \sum_{nj} m_{nm} \alpha_{nj} c_{nj}(t)) + b_{mk}\right]_+ . \qquad (9)
\end{aligned}
$$

This results in

$$
\tau_c \frac{dc_{mk}(t)}{dt} = -\left(c_{mk}(t) - \left[\sum_l \Omega^{int}_{mkl} c_{ml}(t) + \sum_{nj} \Omega^{cpl}_{mnkj} c_{nj}(t) + b_{mk}\right]_+\right) . (10)
$$

The weights $\Omega^{int}_{mkl} = g_{mk}\alpha_{ml}$ define coupling weights internal to a population representing a particular Gaussian coefficient. The weights $\Omega^{cpl}_{mnkj} = \tau_c m_{nm} g_{mk} \alpha_{nj}$, define the coupling between populations representing different coefficients and thus drive the dynamics of the lamprey's swimming. The time constant τ_c defines the time scale of the neuronal dynamics,

Step 5.5.4 Given the nature of this model, we can simply replace *parts* of the Gaussian representation with a neural one. This has great computational advantages, allowing us to simulate one section of the lamprey in great detail (i.e. at the neural level) and concurrently simulate the other sections at a lower level of detail (i.e. the Gaussian level). Our simulations have shown that representing the Gaussian coefficients at the neural level does not adversely affect the model's performance.

3 Strengths of this Model and Method

The main advantages of this model over traditional oscillator models lie in the model's controlability and stability. In contrast to traditional models, we have direct control over the direction and frequency of swimming via the ω parameter (which can also be represented neurally). Adding other parameters is a simple matter of introducing them in the Fourier model and projecting them to the neural space. As well, unlike bistable oscillator models, the stability of this model is guaranteed since the parameters are generated from an analytically stable model. In other words, we explicitly construct a spatial-temporal attractor in neuronal space that meets the criteria of the modeled CPG.

Incorporating more detail (e.g. more realistic muscle dynamics) does not pose an insurmountable challenge or the possibility of unforeseen behaviors using this method. The approach allows us to take into account data available from the neurobiological system. In the case of the lamprey, we could incorporate constraints such as observed connectivity, spatial wave length, and neuron response functions. Additional constraints can be incorporated not *ad hoc*, but by following a procedure similar to the one outlined here.

Perhaps most important is the generality of this approach. Any periodic behavior, including lamprey locomotion, can be thought of as a cyclic attractor in the system's state space. Other attractors, such as line attractors and point attractors, capture other kinds of natural behavior. For example, line attractors provide a good description of the neural integrator (which controls eye position) [4]. These "dynamical systems" descriptions are general ones. From this viewpoint, traditional CPG bistable oscillators are only a special subset of the more general class of attractor motor control circuits. The method presented here is applicable to all such circuits.

References

[1] E. Marder, N. Kopell and K. Sigvardt, How computation aids in understanding biological networks, in: P. Stein, S. Grillner, A. Selverston and D. Stuart, eds., Neurons, networks, and motor behavior (MIT Press, Cambridge, MA, 1997).

[2] E. Marder and R. L. Calabrese, Principles of rhythmic motor pattern generation, Physiological Reviews 76:3 (1996) 687-717.

[3] A. I. Selverston, Are central pattern generators understandable? Behavioral and Brain Sciences 3 (1980) 535-571.

[4] C. Eliasmith and C. H. Anderson, Developing and applying a toolkit from a general neurocomputational framework, Neurocomputing 26 (1999) 1013-1018.

[5] S. Hakimian, C. H. Anderson, W. T. Thach, A PDF model of populations of Purkinje cells, Neurocomputing 26 (1999) 169-175.

[6] T. Wannier, T. G. Deliagina, G. N. Orlovsky, and S. Grillner, Differential effects of the reticulospinal system on locomotion in lamprey, J. Neurophysiology 80 (1998) 103-112.