Original work: C. Eliasmith and C. H. Anderson
Web summary: C. Eliasmith

(also see the related paper)

Introduction

Repetitive behavioral patterns such as swimming, flying, chewing, 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). A CPG is a group of neurons that can produce rhythmic patterns without sensory input. Notably, reciprocally connected model networks with inhibitory weights (i.e. bistable oscillators) are able to produce similar rhythmic patterns. It is often assumed that because such networks produce a behavior similar to that seen in CPGs, they are good models of CPGs. However, the pitfalls of this kind of phenomenological modeling are numerous and severe. For example, Marder et al. have shown that such networks in no way guarantee oscillations and, in fact, equally often produce synchrony: a deadly result for an organism.

Here, we describe an approach in which we determine a desired pattern of output forces and then generate the neuronal control circuits needed to produce that pattern of forces. Traditional CPG bistable oscillators are a special subset of this more general class of motor control circuits. Here we outline this procedure and apply it to construct a novel model of lamprey swimming.

Five Step Process

The iterative five step process we apply in the remainder of this example 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 An(t):

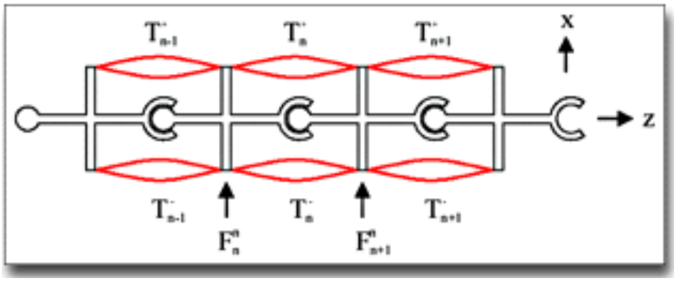
$$\frac{d\mathbf{A}}{dt} = \mathcal{G}\left(\mathbf{A}(t), \mathbf{U}(t), t\right) \qquad (1)$$

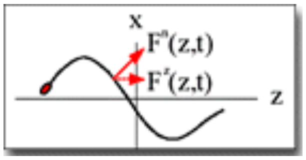
(4) Implement and test the model; and

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

Application to Lamprey Locomotion

Step 1: We can understand the lamprey as a set of finite length connected rods as shown here:





This results in expressions for the forces and tensions in the normal (n) and forward (z) directions (see figure). Thus, we can write the tension as a function of time and lengthwise position as:

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

where $v = \omega * L/2\pi$ is the velocity, L is the length of the lamprey and κ is a scaling parameter. If this equation is satisfied by the muscles along the lamprey's length, they will produce the proper swimming motion for the lamprey (i.e., a traveling wave). Of course, it is the nervous system that controls the muscles. So, assuming a linear muscle plant (a simplification), we expect the motor neurons to be satisfying this same function.

Step 2:

We can now define a standard basis function representation of this function in terms of coefficients (An) and an orthonormal basis (i.e. cos(x) and sin(x)) as follows:

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

where

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

Representing the function in this way makes it amenable to further decomposition into a representation that can be directly implemented in a population of neurons.

Step 3:

In order to implement this representation in a dynamical model, we need to express the An(t) coefficients (i.e., the time-dependent coefficients) dynamically. This results in an equation that is a specific version of equation (1) above:

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

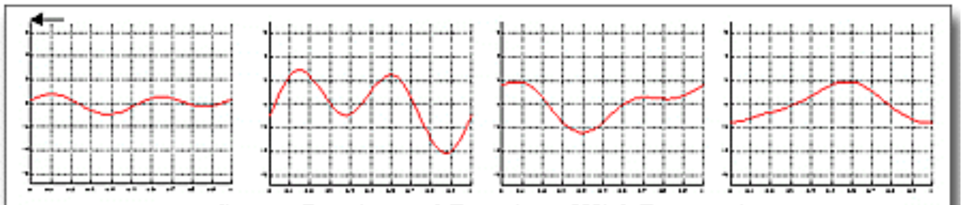
where,

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

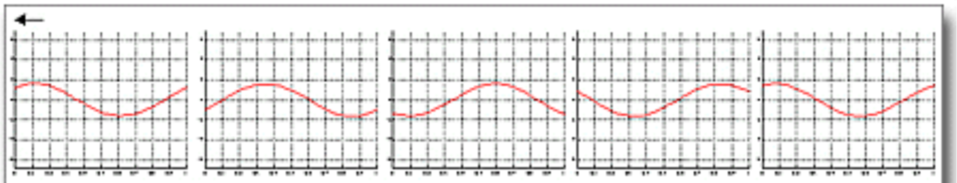
Essentially, we have broken down the transformation matrix M to give us precise control over the dynamics of the model. In particular, Mw controls the steady state oscillatory swimming dynamics; Md uses two rate constants, *a*0 to force the first and third Fourier amplitudes to be equal in magnitude and opposite in sign, and *a* to damp out high spatial frequencies; and Ms controls the start up behavior of the lamprey by inducing exponential growth in the desired amplitudes. Together, these matrices sum to give M, which defines the stepwise dynamics of the system.

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 *w*. As well, high frequency harmonics are damped out and the wave's amplitude increases exponentially to some desired value at startup as controlled by U(t).



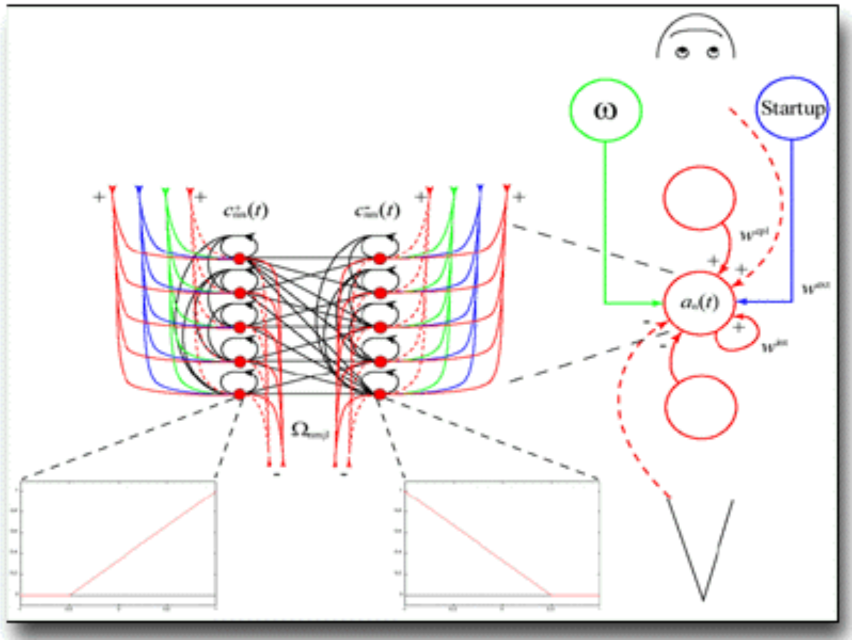
Startup Transient and Damping of High Frequencies



Steady State Swimming

Step 5:

With this simple working model in hand, we can repeat steps 2-4 and project this representation into a progressively more neurologically realistic representation. This will result eventually in a detailed model of lamprey swimming, as diagrammed below.



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 along the length of the lamprey:

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

where am(t) is the amplitude of the mth Gaussian centered at the point zm = m * dz.

We can now construct the projection operator by projecting our previously constructed Fourier basis onto our new Gaussian basis (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 A(t) into dynamical equations in the space of the Gaussian amplitudes a(t). We can write:

$$\frac{d\mathbf{a}(t)}{dt} = \mathbf{m} \cdot \mathbf{a}(t), \text{ where } \mathbf{m} = \Gamma^{-1}\mathbf{M}\Gamma. \quad (2)$$

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

Step 5.4:

We can again simulate the lamprey's swimming. This time the simulation is in the Gaussian space and the dynamical equation is (2). 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 Mw 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, am(t):

$$\begin{aligned} c_{mk} &= \mathcal{F}_{mk}[a_m(t)] && \text{(encode)} \\ a_m(t) &= \sum_k \alpha_{mk} c_{mk}(t) && \text{(decode)} \end{aligned}$$

Here, Fmk[] is the nonlinear spike generation process that *encodes* am(t) into the firing rate of the neuron indexed by m, k. The *a*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, gmk (gain) and bmk (bias), and then finding the optimal weights wmk 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, cmk, 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 and the Gaussian dynamics (2) as follows:

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

This results in

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

The first set of weights (Omega int) define coupling weights internal to a population representing a particular Gaussian coeffcient. The second set of weights (Omega cpl) define the coupling between populations representing different coefficients and thus drive the dynamics of the lamprey's swimming. The time constant 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 the neural level representation successfully implements lamprey swimming.

Strengths of this Model and Method

The main advantages of this model over traditional oscillator models lie in the model's controllability and stability. In contrast to traditional models, we have direct control over the direction and frequency of swimming via the w 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.