# Developing and applying a toolkit from a general neurocomputational framework [1]

Chris Eliasmith [a], 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 63124, cha@shifter.wustl.edu*

**Abstract**

Using a general neurocomputational framework, we develop a set of biologically constrained tools for constructing networks which exhibit interesting behavior. We provide an example of the application of these tools to modeling the part of the brain which controls horizontal eye position. This toolkit and its use is an example of the application of the more general framework.

*Key words:* Representation; Neurocomputation; Line attractor; Neural integrator

## 1 Introduction

Neuroscience has been characterized as field with a dearth of general theories. We see this characterization as a challenge to researchers. Our overriding goal is to meet this challenge by developing a general neurocomputational framework for understanding biological systems. Of course, this is a grand goal and one which cannot be met in the course of a single paper. Here, we take the first few steps in introducing our more general framework; namely, developing a set of simple, yet general, tools and showing how they can be applied to a specific problem. In the first two sections we will develop these tools. In the third section we will show how this toolkit can be employed in constructing a model of the part of the brain responsible for controlling eye movement; the

[1] This research has been supported by the McDonnell Center for Higher Brain Function at Washington University (both authors), the Social Sciences and Humanities Research Council of Canada (Eliasmith) and NFS IBN-9634314 (Anderson).

neural integrator. The neural integrator is intended only as an example of the kind of networks researchers can construct with this toolkit. And, the toolkit itself is only one example of the kinds of tools which can be generated within our general framework.

## 2 Representation and the line attractor

In order to behave appropriately in its environment, an animal needs a means of both transducing and storing information. As well, a successful animal must be able to represent its *desired* behavior. For example, an animal must be able to saccade to something it perceives as a threat. This kind of orienting behavior demands both a successful representation of the position of the threat and a representation of the motor commands necessary to orient the animal's eyes in the appropriate way. Such considerations lead us to take as the fundamental problem of neural computation that of representing (e.g. positions in the world and desired eye positions) and transforming those representations (e.g. from positions in the world to desired eye positions).

A simple yet ubiquitous kind of representation in a biological system is that of an analog value. But we need more than just a representation of an analog value to cope with the real world. In order to be a successful representation it must be dynamically stable; by which we mean that the means of representation should not drastically alter the value of the analog quantity over time. Furthermore, in biological systems, we must assume that the representation into which the value is to be encoded is dependent on some property of neurons. An obvious choice for such a property is the neural response function.

The distinction between the neural representation of eye position and the actual eye position is an important one. We will, after Zemel and Dayan (3), refer to the neural representation as being in the explicit space (i.e. the space of measurable neural firings) and the actual eye position as being in the implicit space (i.e. the space of eye positions measured in degrees). Given this distinction, we are encoding into and decoding from the explicit space. More precisely, we can formulate this relationship as:

$$a_i(\xi) = F[\int \hat{\phi}_i(x) f(x;\xi) dx] \tag{1}$$

$$f(x;\xi) = \sum_i a_i(\xi)\phi_i(x) \tag{2}$$

Where $a_i(\xi)$ are neuron response functions, $f(x;\xi)$ is the parameterized function to be represented in terms of the neuron response functions, $\hat{\phi}_i(x)$ are encoding functions, $\phi_i(x)$ are decoding functions, $F[]$ is some nonlinear operator, and $\xi$ is the implicit space over which the representation is parameterized.

For generality, we perform the encoding and decoding not on a single variable but rather on a function of that variable. The function is parameterized on the implicit space we are interested in representing. In the cases we discuss in this paper, our implicit space is the mean of the function being represented.

Thus, we can write our estimate of the implicit variable, i.e. the mean, as:

$$< \bar{x} >= \int x f(x;\bar{x}) dx = \sum_i a_i(\bar{x}) \int x \phi_i(x) dx \tag{3}$$

Before we can make use of this representation, that is, transform it in various ways, we must determine the encoding and decoding functions. To determine the decoding functions we can construct the following energy:

$$E = \frac{1}{2} \iint [f(x;\bar{x}) - \sum_i a_i(\bar{x}) \phi_i(x)]^2 dx d\bar{x} \tag{4}$$

Minimizing this energy will enable us to find the desired set of optimal decoding functions:

$$V = \Gamma \Phi \tag{5}$$
$$\Gamma^{-} V = \Phi \tag{6}$$
$$\text{Where, } V_j = \int a_j(\bar{x}) f_0(x, \bar{x}) d\bar{x} \tag{7}$$
$$\Gamma_{ij} = \int a_j(\bar{x}) a_i(\bar{x}) d\bar{x} \tag{8}$$

Because $\Gamma$ in this expression is likely to be singular, we must use singular value decomposition to find a pseudo-inverse and solve for $\Phi$.

In order to simplify our analysis, we will assume that the encoding functions, $\hat{\phi}$ are the non-rectified straight lines defined by the neuron response functions. Barber (5) has shown that this is a good approximation to the encoding functions found using gradient descent. The neuron response functions themselves, i.e. $a_i(\bar{x})$, can be determined experimentally though, again, we assume here that they are rectified (i.e. piecewise linear) lines to simplify our analysis. However, the analysis is not at all dependent on this assumption, and similar analyses have been carried out for various neuron response functions and, importantly, spiking neurons (2; 1).

As noted, in order for our representation to be a good one, it must be stable. Substituting (2) into (1), and including a time index we obtain:

$$a_i(\bar{x}(t+\tau)) = \sum_j a_j(\bar{x}(t)) \int \hat{\phi}_i(x)\phi_j(x)dx \tag{9}$$

$$a_i(\bar{x}(t+\tau)) = \sum_j \omega_{ij} a_j(\bar{x}(t)) \tag{10}$$

$$\text{Where, } \omega_{ij} = \int \hat{\phi}_i(x)\phi_j(x)dx \tag{11}$$

This expression is now in the form of a simple recurrent network. Given the representations developed in the previous section, we have a means of directly calculating the weights needed to preserve our representation of the implicit variable over time. This is the same as performing a simple transformation from a value onto itself.

We have found that random initial conditions of such networks relax to a point on the transfer function in approximately one time step. Furthermore, the transfer function approximates a line attractor. That is, at about half of the crossing points of the actual transfer function with the ideal line attractor transfer function, the network exhibits a stable fixed point. The number of such crossings increases linearly with the number of neurons in the network. In effect, we have constructed a network which holds a memory (e.g. analog value) and whose precision depends on the number of neurons in the network [2]. This concludes our derivation of a basic toolkit which can be used to construct more complex networks.

## 3 The neural integrator

In this section we present an application of the computational tools so far developed. In particular, we describe a model of the horizontal neural integrator which controls horizontal eye position given a velocity input. This biological network is found in the rostral medial vestibular nuclei and the nuclei prepositus hypoglossi. The velocity inputs from premotor neurons have been shown to have a background firing rate (6). The output of the neural integrator projects to the motor neurons controlling eye position.

There is evidence that the neural integrator acts much like a line attractor (4). We can express its behavior as:

$$\bar{x}(t+\tau) = \bar{x}(t) + \bar{\tau}_v(t)\bar{v}(t) \tag{12}$$

The integration time step, $\tau_v(t)$, which determines the integration speed of

[2] We have also performed noise sensitivy and RMS error characterizations which show the representation is stable under noise.

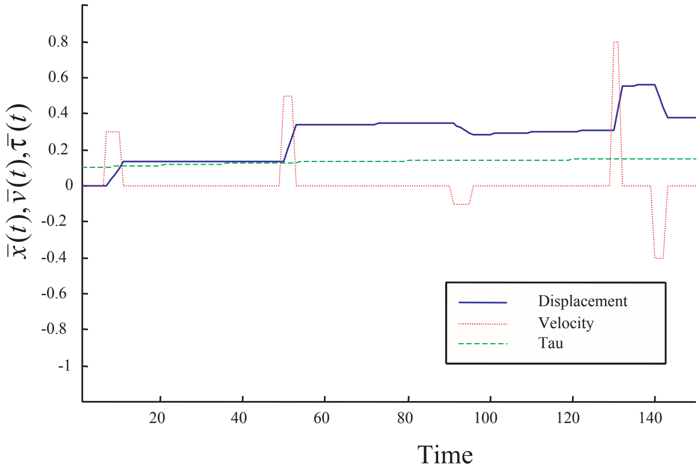


Fig. 1. Time course behavior of the implicit variables in the neural integrator network

the velocity signal can also be stored in a line attractor. Thus, we can construct a network to act as a neural integrator by combining a line attractor storing $\tau_v(t)$, a network representing the velocity command, and a second line attractor which integrates the velocity command and stores the resulting eye position. To calculate the weights for this complex network we can formulate the problem as a calculation over mean values, as suggested by equation (12). Given the representation developed in section 2, we know, because of our straight line encoding functions, that (1) reduces to:

$$a_i(\bar{x}(t+\tau)) = F[\alpha_i \bar{x}(t+\tau) + \beta_i]_+ \tag{13}$$

We can also rewrite (12) as:

$$\begin{aligned} < \bar{x}(t+\tau) > &= < \bar{x}(t) > + < \bar{\tau}_v(t) >< \bar{v}(t) > && (14) \\ &= \sum_j b_j(\bar{x}(t)) x_j + \sum_{lk} d_l(\bar{\tau}_v(t)) \tau_l c_k(\bar{v}(t)) v_k && (15) \end{aligned}$$

Where $b_j$, $d_l$ and $c_k$ are firing rates and $x_j$, $\tau_l$ and $v_k$ are encoding function means, e.g. $x_j = \int x \phi_j(x) dx$, from equation (3). Substituting (15) into (13), we obtain:

$$\begin{aligned} a_i(\bar{x}(t+\tau)) &= F[\sum_j \omega_{ij} x_i + \sum_{lk} \varpi_{ilk} \tau_l v_k + \beta_i]_+ && (16) \\ \text{Where, } \omega_{ij} &= \alpha_i b_j(\bar{x}(t)) && (17) \\ \varpi_{ilk} &= \alpha_i d_l(\bar{\tau}_v(t)) c_k(\bar{v}(t)) && (18) \end{aligned}$$

The network resulting from this derivation has the behavior depicted in figure 1. We can see from this example that the toolkit developed in earlier sections can be applied in a straightforward manner to constructing more complex networks with correspondingly more complex behavior.

## 4 Conclusion

We have provided a toolkit for constructing models which adhere to biological constraints. There are a number of important extensions of these basic tools which merit brief mention. First, this analysis generalizes to any number of dimensions. We have implemented a plane attractor using two-dimensional neuron response functions. Second, the analysis generalizes to any kind of neuron response functions. We have implemented a ring attractor using Gaussian response functions. Third, and perhaps most important, this analysis generalizes to spiking neurons even though the model and analyses presented here assume a rate model (2). In sum, we have demonstrated how basic tools derived from a general framework can be applied to construct networks with biologically relevant behaviors.

## References


[1] C.H. Anderson, Modeling population codes, *Proceedings of CNS *98* (Santa Barbar, CA, July 1998).

[2] Shahin Hakimian, Charles H. Anderson, and W. Thomas Thach, A PDF model of populations of purkinje cells: Non-linear interactions and high variability, *Neurocomputing* **This issue**.

[3] P. Zemel, R. Dayan, and A. Pouget, Probabilistic interpretation of population codes, *Neural Comput.* **10(2)** (1998) 403–430.

[4] H.S. Seung, How the brain keeps the eyes still, *Proc. Natl. Acad. Sci. USA Neurobiology* **93** (Nov. 1996) 13339–13344.

[5] M.J. Barber, Finding encoders and decoders for the PDF framework, (unpublished).

[6] D.M. Waitzman, V.L. Silakov, and B. Cohen, Central mesencephalic reticular formation (cMRF) neurons discharging before and during eye movements, *J. Neurophysiol.* **75(4)** (1996) 1546–1572.