Population models of temporal differentiation

Bryan Tripp and Chris Eliasmith*

July 14, 2009

Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, Ontario, N2L3G1

*To whom correspondence should be addressed; E-mail: celiasmith@uwaterloo.ca

Abstract

Temporal derivatives are computed by a wide variety of neural circuits, but the problem of performing this computation accurately has received little theoretical study. Here we systematically compare the performance of diverse networks, which calculate derivatives using 1) cell-intrinsic adaptation and synaptic depression dynamics, 2) feedforward network dynamics, and 3) recurrent network dynamics. Examples of each type of network are compared by quantifying the errors they introduce into the calculation, and their rejection of high-frequency input noise. This comparison is based on both analytical methods and numerical simulations with spiking leaky-integrate-and-fire (LIF) neurons. Both adapting and feedforward-network circuits provide good performance for signals with frequency bands that are well-matched to the time constants of post-synaptic current decay and adaptation, respectively. The synaptic-depression circuit performs similarly to the depression circuit, although strictly speaking, precisely-linear differentiation based on synaptic depression is not possible, because depression scales synaptic weights multiplicatively. Feedback circuits introduce greater errors than functionally-equivalent feedforward circuits, but they have the useful property that their dynamics are determined by feedback strength. For this reason, these circuits are better suited for calculating the derivatives of signals that evolve on time scales outside the range of

membrane dynamics, and possibly for providing the wide range of timescales needed for precise fractional-order differentiation.

1 Introduction

The brains of behaving animals calculate many derivatives. Most of these arise from spike-rate adaptation (Lundstrom, Higgs, Spain, & Fairhall, 2008) and synaptic depression (Abbott & Regehr, 2004), common cell-intrinsic processes with high-pass, derivative-like dynamics. Notably, both of these processes have been proposed to optimize information transmission (Goldman, Maldonado, & Abbott, 2002; Wark, Lundstrom, & Fairhall, 2007). However, the resulting derivatives may also be used in dynamic computations. For example, a derivative that arises from adaptation and synaptic depression in series (a common occurrence in the cortex) has been proposed to contribute to a predictive computation that compensates for axonal conduction delays (Puccini, Sanchez-Vives, & Compte, 2007).

Notably, despite the prevalence of adaptation and synaptic depression, other derivative computations rely on network mechanisms. For example, a differentiator in the vestibular system, which converts head velocity signals into acceleration signals, is based on the convergence of excitatory inputs and slower inhibitory inputs (Holstein et al., 2004). Excitation combined with delayed inhibition may also drive a differentiation that accounts for retinal direction selectivity (Yoshida et al., 2001). In *C. elegans,* chemotaxis is based on the temporal derivative of chemical attractant concentration, and models combining direct excitation with indirect inhibition through interneurons reproduce this behaviour, and agree with the anatomy (Dunn, Lockery, Pierce-Shimomura, & Conery, 2004). In many other cases, the mechanism of neural differentiation is less clear. For instance, respiratory rhythm is influenced by NMDA-dependent differentiation (Young, Eldridge, & Poon, 2003; Poon, Young, & Siniaia, 2000), but the exact mechanism has not been identified. Similarly, derivative input/output relationships have been observed in human auditory cortex (Jenison, 2007), and in the reflex that stabilizes arterial pressure (Kawada et al., 2004), but the mechanisms are unknown. More abstractly, temporal-difference learning models, which account well for the activity of midbrain dopamine neurons, require an accurate derivative of reward prediction, but there is a lack of consensus about how this computation is performed in the brain (Joel, Niv, & Ruppin, 2002).

In addition to the specific, well-documented instances above, differentiation is likely to play a broader role in motor control, as suggested by the importance of differentiation in engineering control theory. Despite the prevalence of state-space methods in modern control theory, output feedback remains an important method by which engineers ensure robustness, either to uncertainty in the plant or to changes in the plant over time (in neuromuscular control such changes can occur due to muscular fatigue). The inclusion of a derivative component in output feedback reduces overshoot and oscillation around the endpoint of a movement. Interestingly, this type of movement error is observed in cerebellar ataxia, and can be induced by transcranial magnetic stimulation of motor cortex (Topka et al., 1999). Furthermore, derivative feedback allows higher-gain proportional feedback, indirectly improving control precision (Vegte, 1994).

1.1 The problem of differentiation

Despite the prevalence of derivatives in neural systems, and their interest as a fundamental dynamic operation, factors affecting the accuracy of these computations have received little theoretical study. In contrast, neural mechanisms of temporal integration have received much attention (Robinson, 1989, 1968), especially for their role in oculomotor control (Seung, 1996) and working memory (Goldman, Levine, Major, Tank, & Seung, 2003; Miller, Brody, Romo, & Wang, 2003). Theoretical studies of neural integration have highlighted important questions regarding their stability: integrator circuits that incorporate realistic neuron models tend to drift toward attractors that arise from imperfect weight tuning (Koulakov, Raghavachari, Kepecs, & Lisman, 2002). Here, we undertake an analogous theoretical study of neural differentiation.

The main barriers to accurate neural differentiation are the minimization of distortion and the rejection high-frequency noise. Distortion errors in neural circuits are loosely analogous to quantization errors in digital systems, in that they arise because signals are approximated through the organization of relatively generic devices, resulting in small residual differences between the ideal and approximate representations. This is potentially a serious problem for differentiation, because differentiating fundamentally involves the difference between an input signal and some internal state that slightly lags the input. (For example, in synaptic depression, the size of the readily-releasable pool of synaptic vesicles lags spiking activity.) The difference at any moment in time between the lagged and non-lagged signals will typically be small, and distortions in these signals can easily be of the same order of magnitude, creating fluctuations that obscure the derivative.

The second barrier to accurate differentiation is high frequency noise. Noise arises from many sources in neural systems. As well, frequencies much higher than those of overt behavioural relevance abound in neural circuits, because communication between neurons is mediated by temporally sharp (and hence spectrally broad) action potentials. Unfortunately, differentiation naturally amplifies high frequencies (by the chain rule, i.e. $\frac{d}{dt}sin(\omega t) = \omega cos(\omega t)$). Therefore, practical differentiators must be band-pass filters: like pure differentiators, they must amplify signal-band inputs in proportion to their frequency, but they must respond minimally to noise components at higher frequencies.

Our present goal is to advance the basic theory of temporal differentiation in neural systems, by contrasting different classes of differentiator networks in terms of these two sources of error. To this end, we present three types of neural networks that perform temporal differentiation in different ways, and analyse examples of each type in terms of distortion error and rejection of high-frequency noise. These example circuits are archetypes of different differentiator categories, rather than models of specific, identified differentiator circuitry, but the results presented below provide a sound theoretical context for more specific models.

1.2 Relationship with past work

A number of specific differentiator networks have been modelled in the past (Holstein et al., 2004; Kawada et al., 2004; Young et al., 2003; Poon et al., 2000; Dunn et al., 2004). One previous study (Puccini et al., 2007) also performed a frequency-response analysis similar to the one presented here. The present study extends this approach to several types of differentiator networks. It also models error propagation through these networks, in such a way that their performance can be compared analytically. As far as we know, this type of systematic, analytical comparison of a wide range of neural differentiator mechanisms has not been undertaken previously. The scope of this study is limited to integer-order differentiation. However, the analytical approach, and most of the models, extend in a straightforward manner to fractional-order differentiation (see Discussion).

2 Differentiator Networks

We begin by developing abstract dynamical models of each type of differentiator, and then elaborate these abstract models into population-coding networks of spiking neurons.

As noted above, we use the term "differentiator" to mean "band-pass filter", which is a practical approximation of an ideal differentiator, in the presence of noise. The required characteristics are easily described in the Laplace domain. In the Laplace domain, the derivative of a signal u is su, where s is the complex Laplace variable, which increases proportionally with signal frequency. A system acting as a practical differentiator should have a transfer function similar to $y/u = s/(\tau s+1)$, where y is the output of the system and τ is a small constant. (Note that this is the derivative of $y/u = 1/(\tau s+1)$, which is the Laplace transform of the non-amplifying impulse response $(1/\tau)e^{-t/\tau}$, an exponential function with integral one.) For low-frequency inputs (small s), such a system will output essentially the derivative $y = su$ (assuming zero initial conditions), while for higher frequency inputs (large s) the output will be roughly proportional to the input. Such a transfer function is therefore high-pass. A transfer function with a higher-order denominator behaves similarly, but further reduces the response to high-frequencies, becoming band-pass.

We study three types of networks that have the required dynamic properties: 1) networks with fast excitation in parallel with slower inhibition; 2) networks with neurons that have intrinsic habituating behavior; and 3) networks with feedback that produces band-pass dynamics. Note that we do not study any examples of single-cell differentiator models. This is because considering intra-cellular mechanisms in a network context allows us to compare these mechanisms more directly with network mechanisms, and because in any case, mammalian neurons typically operate in large correlated groups. Sketches of examples of each type of network are shown in Figure 1. In this figure, each circle indicates an ensemble of neurons that encodes a scalar value, in a population code. Each of these networks is described in more detail in section 2.2.

2.1 Abstract Models vs. Detailed Network Models

Abstract dynamical models of each of these network structures are shown in Figure 2, as Laplace-domain block diagrams. We will first briefly discuss the abstract models themselves, before going into detail about their relationship with neurons (see paragraph after next and Figure 3).

To obtain the abstract models, it is assumed that the firing activity of each ensemble of neurons encodes a scalar variable. For example, the ensemble labelled "input" in Figure 1A encodes the instantaneous value of the signal that is to be differentiated. A node is created in the block diagram for each encoded variable. We then add dynamic blocks, which contain the Laplace variable s, to model the relevant neural dynamics. For neurons that fire at a constant rate in response to constant input, feed-forward dynamics are dominated by post-synaptic current dynamics (Eliasmith & Anderson, 2003), and the transfer function of an ensemble closely approximates the transfer function of the post-synaptic current. These post-synaptic current dynamics are approximated as single-time-constant exponential decays, with

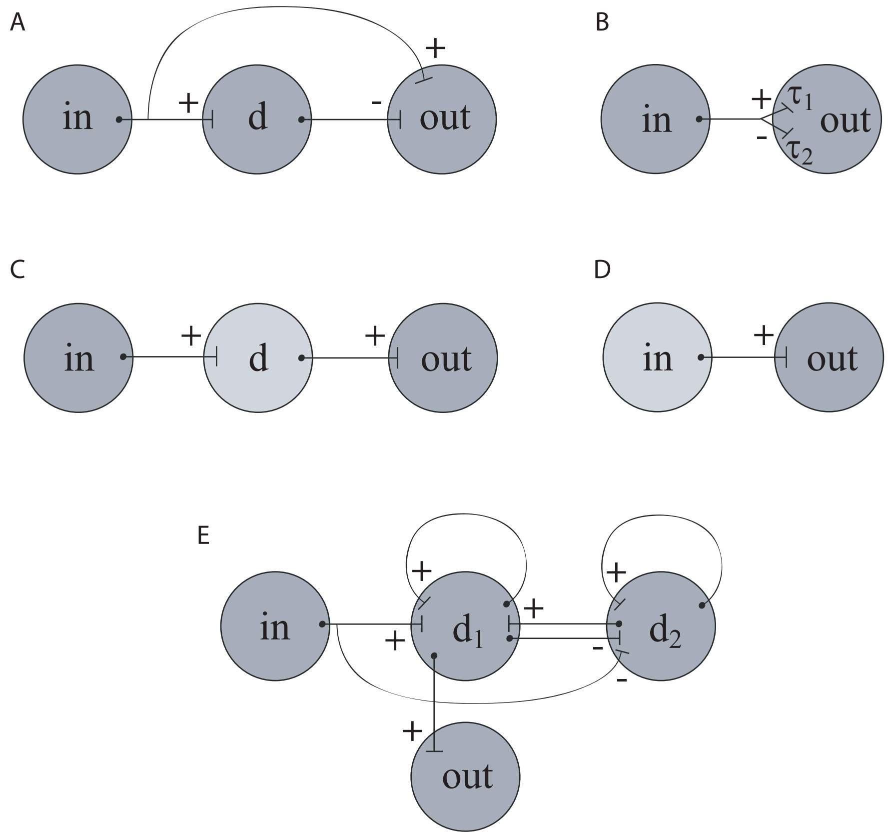

Figure 1: Structures of example differentiator networks. Each circle indicates a population of neurons, and each line indicates an axonal projection from one population to another (the flat end indicates axon terminals). Populations are labelled according to the information that they encode, i.e. input to be differentiated (“in”), differentiated output (“out”), and state variables used in the calculation of the derivative (“d”). For now, plus (+) and minus (-) signs can be taken to indicate excitation and inhibition, respectively, but a more sophisticated mapping to addition and subtraction of encoded variables will be defined shortly (section 3). A and B show examples of feedforward networks. In A, inhibition is slowed by passage through an intermediate population. In B, inhibition is slowed relative to excitation because the post-synaptic current time constant τ_2 is longer than τ_1. C, Structure of a differentiator network based on cell-intrinsic firing-rate adaptation. Here the ensemble “d” consists of adapting neurons. D, Differentiator based on short-term depression in synapses that originate from the ensemble labelled “in”. E, A feedback network structure with two populations that represent state variables. This structure can support a wide range of 2nd-order dynamics, of which two examples will be presented.

impulse response $\frac{1}{\tau}e^{-t/\tau}$ (for $t > 0$) and Laplace transform $1/(\tau s + 1)$. Figure 2C and D also contain dynamic blocks that model firing-rate adaptation and short-term synaptic depression, respectively. These dynamics are also modelled as single-time-constant exponential decays, with different time constants. This is a simplification, in that these processes operate on multiple timescales, which can lead to approximately fractional-order dynamics over large frequency ranges (we return to this issue in the Discussion). Finally, we add static blocks, which correspond to mappings between encoded variables that arise from synaptic weights. These static mappings are parameterized to perform whatever amplification is required, so that the output of the network is the derivative of the input. The block diagram for each network is described in detail in Section 2.2.

Each of these abstract dynamical models serves as a starting point for a network model composed of leaky-integrate-and-fire (LIF) neurons. In these models, the dynamics of adaptation and synaptic depression are modeled in more detail, as described in the following section. Post-synaptic current dynamics are again modeled as first-order filters. Within each ensemble, neuron parameters are varied randomly across neurons, so that a given input to an ensemble elicits a variety of neuron responses. This type of response variation allows an ensemble that encodes a scalar x to project a function $f(x)$ to post-synaptic ensembles, where $f(x)$ depends on synaptic weights. Synaptic weights are chosen to optimally approximate the transforms specified in the abstract models, using the methods described in the Appendix (see also Eliasmith & Anderson, 2003). These detailed circuit models are therefore closely related to the corresponding abstract models, because: 1) the abstract and detailed models have similar dynamic elements; and 2) the synaptic weights in the detailed models are optimized to approximate the static blocks in the abstract models. In other words, each network model is an optimal approximation of a certain idealized differentiator, given (modelled) biophysical constraints. The relationships between block diagrams and network models is illustrated in Figure 3. As we verify below (Sections 3 and 4), the detailed models behave very similarly to the abstract models.

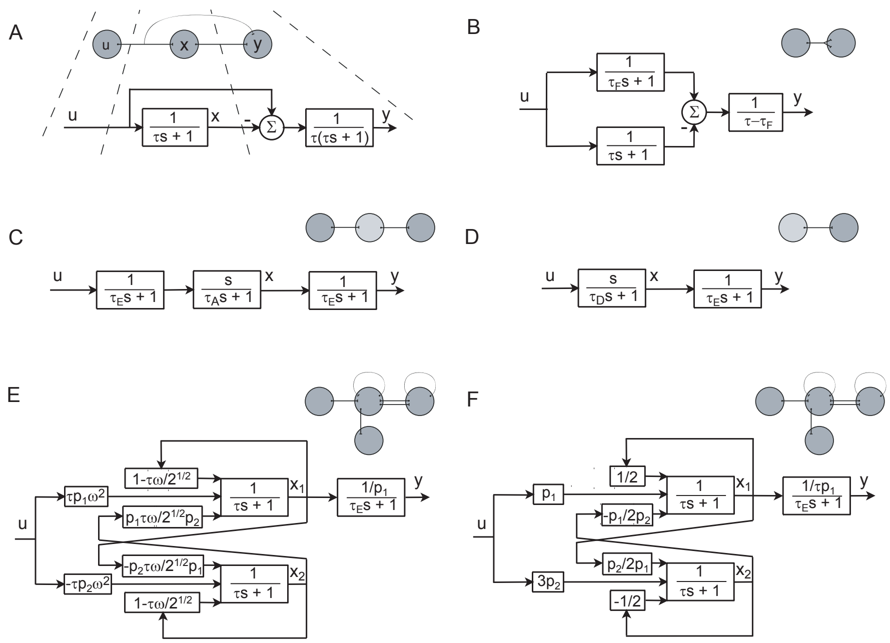

Figure 2: Block diagram models of the neural network structures in Figure 1 (insets show the corresponding network structures; correspondences are indicated by dashed lines in A). Each model has an input u and an output y. Each block corresponds to a transfer function. Those containing the Laplace variable s are dynamic transfer functions, and those that do not contain s are static functions that simply scale their inputs. The variables x_i are internal state variables. The dynamic blocks are parameterized with time constants: τ (without a subscript) is the time constant of post-synaptic current decay in all neuron populations, unless otherwise specified; τ_F is the time constant of the fast-decaying post-synaptic currents in the dual-time-constant network; τ_A and τ_D are time constants of adaptation and synaptic depression, respectively, which are defined in the text (section 2.2.2); and τ_E is the time constant of synapses external to the differentiating part of the network. In panel E, ω is a corner frequency, and p_i are scale factors, which are also defined in the text (section 2.2.3). A, intermediate-ensemble network; B, dual-time-constant network; C, adaptation network; D, synaptic-depression network; E, butterworth network; F, second feedback network that approximates intermediate-ensemble network.

This optimization allows us to study the best possible performance of different circuit architectures. The resulting functional similarity between detailed and abstract models also allows us to use the abstract models for analysis. The abstract models are of interest because 1) they are readily analysed, and 2) each one concisely describes a large family of nearly-equivalent detailed models, since in a detailed network model, a change in neuron parameters can be largely offset by a change in synaptic weights. Networks thus related will behave very similarly, although there will probably be differences in the accuracy of their population codes. The effects of these subtle differences on network performance could be determined by sampling randomly from the family of detailed models that correspond to a single abstract model. However, we instead elaborate the abstract models with explicit models of coding errors (see section 5). This approach allows us to perform parametric analyses that concisely describe entire network families.

It should be noted that because differentiation is a linear operation, all of the networks must have essentially linear responses over their operating ranges. There are many non-linearities in the detailed network models. However, because the synaptic weights are optimized to approximate the linear abstract models, using the methods described in the Appendix, these non-linearities largely cancel each other out. For example, if one neuron were to saturate around a certain input value, and another were to begin firing around the same input value, then summing these responses with appropriate weights would produce a net response that was more linear than either neuron's response alone. The optimization method produces quite linear net responses from hundreds of such combinations. This is possible in general because as long as the nonlinear neuron responses are sufficiently diverse, their first few principal components typically span something very close to a linear function. The result is that the detailed models have near-linear input-output maps, despite many local non-linearities.

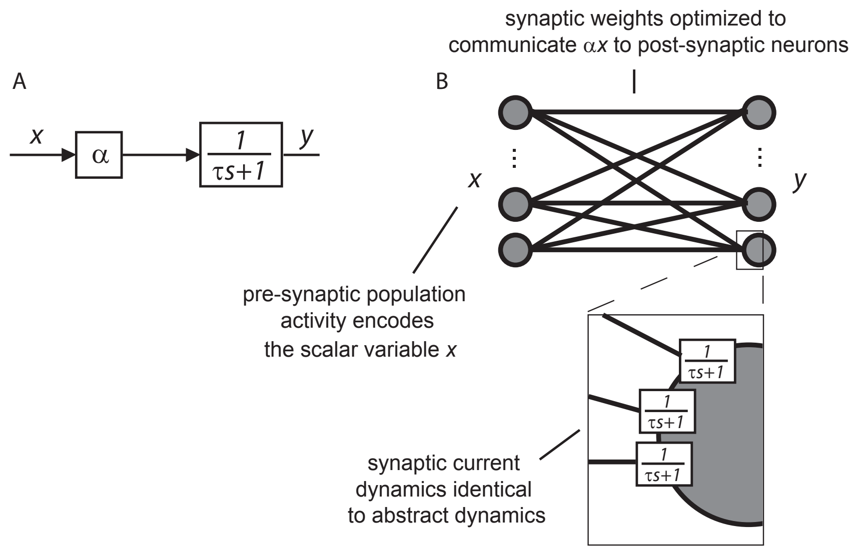

Figure 3: Relationship between abstract and detailed models. A, Example block diagram with two state variables (x and y). The diagram also includes a static block, which multiplies x by α, and a dynamic block, which filters αx to produce the output y. B, Schematic of a neural network model that corresponds to this block diagram. Each circle indicates a neuron. On the left is a population of neurons that encodes the value of x (i.e. the firing of these neurons is correlated with x, in such a way that x can be estimated from their activity pattern). These neurons project to another population of neurons on the right, which encodes the value of y. The synaptic weights in this projection are chosen so that an optimal estimate of αx is conveyed to the y neurons (the procedure for determining these synaptic weights is described in the Appendix). The transfer function of each synapse, which maps a spike input to a post-synaptic current response, is the same as the transfer function of the block diagram in A. As a result of these similarities, the population-coding network essentially exhibits the behavior described by the block diagram.

2.2 Types of Differentiator Networks

At noted above, for any block-diagram model of a neural circuit, there are infinitely many detailed models that behave in essentially the same way. However, even at the block diagram level, there are many possible variations on each basic network type. For example, consider the type of differentiator that compares a fast excitatory signal with a slower inhibitory one. An example of this type is sketched in Figure 1B. In this example, the inhibitory projection is slower because the inhibitory post-synaptic dynamics have a longer time constant. However, inhibition could also be slowed by passage through an intermediate ensemble (as in Figure 1A) or by a greater pure delay in conductance. In short, there are many possible combinations of network topology and cellular dynamics that may implement any of the three types of differentiators described above. To avoid excessive length, only two specific examples of each type of circuit are discussed (the chosen examples are attractive because they are relatively simple and obvious, but there is no definite justification for these choices). While this makes the analysis non-exhaustive, the chosen examples reflect simple, biologically-relevant strategies, so we expect that many of the unconsidered possibilities are variations on these themes.

2.2.1 Parallel Feedforward Networks

We begin with networks that calculate the derivative of an input signal using a fast feedforward excitatory projection, in parallel with a lagged inhibitory projection. Specific examples differ in the ways in which the inhibitory projection is lagged.

In the first example, the input ensemble projects directly to the output ensemble (no lag), and also indirectly through another ensemble, which introduces a lag. This structure is shown in Figure 1A: a projection from the input ensemble (labelled “in”) branches to both the output ensemble (labelled “out”) and an intermediate ensemble (labelled “d”). The intermediate neurons in turn project to the output neurons. This form of circuit has been suggested to underlie chemotaxis in *C. elegans* (Dunn et al., 2004). Figure 2A shows the corresponding block diagram. In this panel, the input u corresponds to the scalar variable that is represented by the input ensemble. The two branches along which u travels correspond to the two projections from the "in" ensemble in Figure 1A. At the end of each projection, the signal enters a 1^{st}-order filter that models post-synaptic current dynamics. This filtering models the communication of the represented variable to a post-synaptic ensemble through synaptic connections, as described in the Appendix. The variable x on the diagram corresponds to the scalar variable that is represented by the intermediate ensemble. The negative of this variable is projected to the output ensemble, where it sums with the direct projection of u. This subtraction is critical to the computation. As discussed in Section 3 (below) there are a number of biophysical mechanisms by which the abstract subtraction operation might be realized through inhibitory synapses; detailed network models will be developed assuming the mechanism described by Parisien et al. (2008). Finally, the variable $y \approx su$ at the right of the diagram corresponds to the scalar variable that is represented by the output ensemble.

In the second circuit (Figure 2B), two copies of the input signal are projected directly and simultaneously onto the output ensemble, but post-synaptic current dynamics are slower in one projection (bottom branch in the diagram) than in the other. The fast time constant is modeled on AMPA receptors ($\tau_1 = .005s$), and the slow time constant on $\mathrm{GABA_B}$ receptors ($\tau_2 = .1s$).

It should be noted that the vestibular differentiator (Holstein et al., 2004) has either this form, or a closely related form that exploits a difference in conductance delays rather than in time constants. In theory, it would be possible for a circuit to combine both conductance delay and time constant differences. However, the transfer function of such a circuit would contain an additional, non-differentiating term, unless either the slow dynamics were much slower than the fast dynamics, or of higher order.

2.2.2 Networks with Cell-Intrinsic Habituating Behavior

The second type of differentiator circuit is based on habituating processes within neurons, which combine to produce band-pass behavior at the population level. We present examples based on firing-rate adaptation (Figure 2C) and synaptic depression (Figure 2D).

Neurons with firing-rate adaptation are modeled as adapting leaky-integrate-and-fire (ALIF) neurons. This model provides a good approximation of the adaptation effects observed in more detailed compartmental models (Koch, 1999; Naud, Berger, Badel, Roth, & Gerstner, 2008). We use an implementation (Camera, Rauch, Lüscher, Senn, & Fusi, 2004) in which adaptation is driven by an unspecified chemical species N, the concentration of which varies as

$$d[\mathrm{N}]/dt = -[\mathrm{N}]/\tau_N + A_N \sum_k \delta(t - t_k).$$

Here [N] is the concentration of the chemical species responsible for adaptation, τ_N is a time constant of decay of [N], and $A_N > 0$ is an increment in [N] with each spike. This model can also be expressed in terms of firing rates (rather than spikes) as

$$d[\mathrm{N}]/dt = -[\mathrm{N}]/\tau_N + A_N r(u, [N])$$

where $r(u)$ is the firing rate. In the present model, these neurons are parameterized so that their unadapted firing rates vary nearly linearly with driving current. Considering the high-pass dynamics of adaptation in conjunction with the low-pass dynamics of synapses, each neuron then behaves like a linear band-pass filter with a negative zero (i.e., firing rate is a sum of low-pass and band-pass components). If γ is the derivative of the unadapted firing rate with respect to driving current, then as long as the neuron's firing rate does not drop to zero,

$$r(u, [N]) = \gamma(I_d(u) - g_N[N]),$$

where $I_d(u)$ is the difference between the net driving current and current threshold for firing, and g_N scales the adaptation-related current. Then the neuron will have first-order dynamics with time constant

$$\tau_A = (1/\tau_N + \gamma g_N A_N)^{-1},$$

(Liu & Wang, 2001). Summarizing, the driving current I_d is mapped to the weighted output y_i at the i^{th} synapse by the state equations

$$d[\mathrm{N}]/dt = -(\frac{1}{\tau_{\mathrm{N}}} + \gamma \mathrm{g_N A_N})[\mathrm{N}] + A_N \gamma I_d,$$

$$y_i = w\gamma(I_d - g_N[N]),$$

where w is the synaptic weight, yielding the transfer function

$$\frac{y_i(s)}{I_d(s)} = w\gamma\tau_A \frac{s + 1/\tau_N}{\tau_A s + 1}.$$

For each neuron, we set $g_N = 1$, and γ between 35 and 45 spikes/s per unit of current (arbitrary units). The remaining adaptation parameters, A_N and τ_N, are set so that 1) $\tau_A =$ .1, and 2) the neuron adapts as strongly as possible, without adapting so much that its firing rate drops to zero with a sudden drop in synaptic drive. Specifically, $\tau_N = \tau_A(\beta/\alpha + 1)/2$, where β is the neuron's bias current (see Appendix), α is the net synaptic current flowing into the neuron per unit of the encoded scalar variable (which is normalized to the range -1 to 1), and $A_N = (1/\tau_A - 1/\tau_N)/\gamma$. Within an ensemble, 75% of the neurons are adapting. The other (non-adapting) neurons perform a static input-output mapping that cancels out the zero of the adapting neurons (which arises from the term $1/\tau_N$ in the numerator, above). The ensemble output is scaled through appropriate synaptic weights, to produce the net ensemble dynamics,

$$\frac{y(s)}{u(s)} = \frac{s}{\tau_A s + 1}.$$

The second example of this type of network (Figure 2D) is based on short-term synaptic depression. Synaptic depression is usually a pre-synaptic phenomenon that arises from depletion of the readily-releasable pool of synaptic vesicles. In simple models, the readily-releasable pool is depleted sharply with each spike and replenished with exponential dynamics (Zucker & Regehr, 2002). The dynamics of the readily-releasable pool are therefore closely analogous to those of [N] (above). Let F be the proportion (between 0 and 1) of the pool that is depleted with each spike, and let $S(t)$ be the remaining proportion of the pool at time t. If the synaptic weight with a full readily-releasable pool is w, then the effective synaptic weight at any instant is $wS(t)$. If we make the same assumptions as in the adaptation model, i.e., that the firing rate $r > 0$ and the response function is linear in this range (i.e. $r(u) = \gamma I_d(u)$), then the state equations relating input current I_d to weighted output y_i are

$$dS/dt = (1 - S)/\tau_S - FS\gamma I_d,$$

$$y_i = wS\gamma I_d,$$

where τ_S is the time constant with which the readily-releasable pool is replenished, and γ is again the derivative of the firing rate with respect to the driving current I_d.

Importantly, despite many similarities with adaptation dynamics, both the dynamic and output equations are non-linear in the depression case, because each equation contains a product of the input I_d and the state variable S (unlike the adaptation model above, which is linear). The dynamic nonlinearity is weak when F is low, and it is also possible that more elaborate models might further linearize the dynamics. However, the output nonlinearity is unavoidable, and it implies that synaptic depression can only produce linear differentiating behavior when the temporal derivative of the firing rate is zero. Thus the intuitive notion that synaptic depression produces derivative-like output is necessarily an approximation. In other words, in contrast with the other models studied here, which exhibit random errors around ideal linear behavior, this network exhibits random errors around behaviour that is systematically non-ideal, in that it is mildly nonlinear.

It must be noted that this stark contrast between linear adaptation and nonlinear depression is exaggerated by the simplicity of the models (all of which ignore many nonlinear properties of real neurons), and in particular that more detailed adaptation models contain nonlinearities. The main nonlinearity that has been neglected in our adaptation model arises from the dependence of mean membrane potential on the firing rate (Benda & Herz, 2003). However, this nonlinearity is arguably less essential to adaptation than the above multiplicative nonlinearity is to depression, because it is small relative to the linear component (Benda & Herz, 2003). Furthermore, the nonlinear component depends on a number of variable factors that could, conceivably, interact to bring it close to zero in some cases. For example, it is increased when the membrane potential approaches spike threshold slowly at low firing rates, suggesting a dependence on noise parameters (due to early noise-driven threshold crossings). To summarize, it is certainly a simplification to model adaptation as linear, but this simplification is reasonable in our view, whereas we do not see analogous arguments for making the same simplification in the case of synaptic depression.

Nevertheless, the depression equations can be linearized about the operating point (S_0, I_0), corresponding to the neuron's mean input and the associated steady-state S, as

$$\frac{d}{dt}\delta S = -(\frac{1}{\tau_S} + F\gamma I_0)\delta S - FS_0\gamma\delta I,$$

$$\delta y_i = w\gamma I_0\delta S + wS_0\gamma\delta I,$$

where $\delta S = S - S_0$, $\delta I = I_d - I_0$, and $\delta y_i = y_i - y_{i0}$.

To facilitate comparison with the adapting-neuron network, depression parameters are set so that the linearized model of depression decays with time constant $\tau_D = .1s$. Specifically, $F = 1/(2\tau_D\gamma I_0)$, where γI_0 is the neuron's mean firing rate, and $\tau_S = 2\tau_D$. These parameters result in weak depression. Weaker depression makes the output more linear, but it also makes the derivative-like output signal smaller relative to the noise. With these parameters, the linearized model has transfer function

$$\frac{\delta y_i(s)}{\delta I(s)} = w\gamma S_0\frac{\tau_D s + \frac{1}{2}}{\tau_D s + 1}.$$

Analogously to the adapting model, parallel non-depressing synapses are weighted to cancel out the zero, and depressing synaptic weights are chosen to yield the net ensemble dynamics,

$$\frac{y(s)}{u(s)} \approx \frac{s}{\tau_D s + 1}.$$

To reiterate, this transfer function approximates the nonlinear dynamics that arise from the synaptic-depression model above, whereas the transfer function of the adapting-neuron ensemble corresponds directly to the simplified adaptation model.

Note that our convention is for each network to contain one ensemble with firing rates that encode the input signal, and another ensemble with firing rates that encode its approximate derivative. The adaptation network therefore requires three ensembles (input, adapting, and output), because the firing rates of the adapting neurons correspond to neither the input nor its derivative, while the depression network requires only two ensembles (input and output). Consequently, the depression network's block diagram contains one fewer post-synaptic dynamics block.

2.2.3 Feedback Networks

The final type of differentiator we explore is again based on network properties, but on feedback rather than feedforward dynamics. There are many different feedback circuits with band-pass behavior. The scope of this diversity becomes clear if we consider that for any set of explicit ordinary differential equations, there is a family of neural circuits with essentially the same dynamics (Eliasmith, 2004). As an interesting consequence, a feedback network model can be parameterized to match observed differentiator dynamics. Two second-order band-pass networks are presented as examples. The first of these (Figure 2E) acts as a Butterworth filter (Sedra & Smith, 1991). This is a standard engineering filter, which maximizes stop-band attenuation without introducing ripples into the frequency response (this idealized filter is chosen to emphasize the versatility of the feedback network structure). The second example (Figure 2F) has the same transfer function as the interneuron circuit (section 2.2.1), in order to provide a point of comparison between feedforward and feedback circuits.

The transfer function of the 2nd order band-pass Butterworth filter is:

$$\frac{y(s)}{u(s)} = \frac{\omega^2 s}{s^2 + 2^{\frac{1}{2}}\omega s + \omega^2},$$

where ω is the corner frequency, in radians/s. As is the case with any linear system, there are infinitely many ways to express this transfer function as a system of ordinary differential equations. In theory, it should be possible to find the realization that optimizes performance in the face of population coding errors and noise. However, this problem is not straightforward, and is outside the present scope. Instead, we begin with a canonical realization, and adjust it in order to make good use of the neurons' operating ranges. The resulting equations are:

$$\dot{x_1}(t) = \frac{1}{\sqrt{2}}(-\omega x_1(t) + \frac{p_1}{p_2}\omega x_2(t)) + p_1\omega^2 u(t),$$

$$\dot{x_2}(t) = \frac{1}{\sqrt{2}}(-\frac{p_2}{p_1}\omega x_1(t) - \omega x_2(t)) - p_2\omega^2 u(t),$$

$$y(t) = x_1(t)/p_1.$$

The variables p_1 and p_2 scale the two state variables to between -1 and 1, within which range the firing rates of the neurons that encode the state variables do not saturate. To find a family of neural circuits with these dynamics, we transform the above state space model into a form that uses post-synaptic current dynamics in place of integration (Eliasmith & Anderson, 2003). The resulting neural circuits have a form that is similar to the state-space model, but with stronger feedback. We begin by writing the dynamic equations above in the Laplace domain, using vector and matrix notation. This gives

$$s\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} -\omega & p_1\omega/p_2 \\ -p_2\omega/p_1 & -\omega \end{bmatrix}\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} + \begin{bmatrix} p_1\omega^2 \\ -p_2\omega^2 \end{bmatrix} u(s),$$

where s is the Laplace variable. It is then possible to rewrite this equation in terms of first-order post-synaptic current (PSC) dynamics, which have transfer function $1/(\tau s + 1)$, as

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = \frac{1}{(\tau s+1)} \begin{bmatrix} 1-\tau\omega/\sqrt{2} & \frac{p_1}{p_2}\tau\omega/\sqrt{2} \\ -\frac{p_2}{p_1}\tau\omega/\sqrt{2} & 1-\tau\omega/\sqrt{2} \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} + \frac{\tau\omega^2}{(\tau s+1)} \begin{bmatrix} p_1 \\ -p_2 \end{bmatrix} u(s).$$

$$y(s) = \begin{bmatrix} 1/p_1 & 0 \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix}.$$

These equations lead directly to the structure shown in Figure 2E. For example, in the block diagram of Figure 2E, there are three inputs to the x_1 node, all of which pass through the block $1/(\tau s + 1)$. One of these inputs is from u, another feeds back from x_1, and the third is from x_2. These three inputs correspond to the three terms on the right-hand side of the equation for x_1, above, all of which share the term $1/(\tau s+1)$. The differentiator ensembles d_1 and d_2 encode the state variables x_1 and x_2, respectively. Note that the dynamic behaviour of this circuit as a whole is not governed only by τ, but that it depends also on the strength of feedback connections. This contrasts with the feedforward networks above, in which dynamics are determined by τ.

For consistency with other models, the output of the Butterworth filter feeds into an output ensemble, which represents the derivative. The signal therefore passes through additional post-synaptic current dynamics, yielding the full transfer function,

$$\frac{y(s)}{u(s)} = \frac{\omega^2 s}{(s^2 + 2^{\frac{1}{2}}\omega s + \omega^2)(\tau_E s + 1)},$$

where τ_E is the post-synaptic time constant of the synapse onto the output ensemble (the subscript “E” stands for “external”, as in external to the core network that performs differentiation).

The second feedback network (Figure 2F) has transfer function

$$\frac{y(s)}{u(s)} = \frac{s}{(\tau^2 s^2 + 2\tau s + 1)(\tau_E s + 1)},$$

which is identical to the feedforward circuit with an intermediate ensemble (Figure 2A) except for the additional post-synaptic current dynamics associated with the output ensemble. Here τ is the time constant of the post-synaptic current dynamics taken from the circuit in Figure 2A. This transfer function is realized as

$$s \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = \frac{1}{2\tau} \begin{bmatrix} -1 & -p_1/p_2 \\ p_2/p_1 & -3 \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} + \frac{1}{\tau} \begin{bmatrix} p_1 \\ 3p_2 \end{bmatrix} u(s),$$

$$y(s) = \begin{bmatrix} 1/\tau p_1 & 0 \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix},$$

where again p_1 and p_2 are scale factors. Like the Butterworth filter, these equations can be re-written in terms of post-synaptic current dynamics, as

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = \frac{1}{2(\tau s + 1)} \begin{bmatrix} 1 & -p_1/p_2 \\ p_2/p_1 & -1 \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} + \begin{bmatrix} p_1 \\ 3p_2 \end{bmatrix} u(s),$$

$$y(s) = \begin{bmatrix} 1/\tau p_1 & 0 \end{bmatrix} \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix},$$

which leads to the abstract model shown in Figure 2F. Note that in this model the time constant of the network equals the time constant of post-synaptic current (i.e. τ). This causes some τ to cancel out, making these equations relatively simple.

3 Simulations

This section presents numerical simulations with networks of spiking neuron models, so that their behavior can be compared with that of the abstract models. Simulation code is available from the authors' web site (http://compneuro.uwaterloo.ca).

As discussed in the previous section, the structure and synaptic weights of the full spiking models are derived systematically from the abstract dynamical models. However, this procedure leaves a number of parameters to be specified, including the number of neurons in each ensemble, and parameters of the spiking neuron models. Each of the simulated circuits contains 2000 input neurons and 1000 output neurons. Additionally, all but the dual-lag and synaptic-depression circuits contain 2000 differentiator neurons. Additional neuron parameters are given in the Appendix.

We must also specify how excitation and inhibition interact to produce increases and decreases in encoded variables. So far, we have assumed that excitation of neurons increases the value of the variable encoded by the population, and inhibition decreases this value. However, in a population-coding network, there are a number of other possibilities. The main constraint is that the outgoing synaptic weights of any given neuron should all have the same sign, because individual neurons are usually either excitatory or inhibitory, except in rare cases.

To satisfy this constraint, we choose a structure that is typical of the cortex, in which excitatory projection neurons synapse directly onto post-synaptic targets, and also indirectly through a smaller group of inhibitory interneurons. This combination of direct excitation and indirect inhibition can closely approximate any set of mixed excitatory and inhibitory synaptic weights, as discussed extensively elsewhere (Parisien, Anderson, & Eliasmith, 2008). This allows us to first specify an approximate model, with projections that consist of optimal unconstrained synaptic weights, and then to transform each of these idealized projections into physiologically realistic projections that function in essentially the same way. The transformation to this form (Figure 4) involves biasing the original synaptic weights, which have

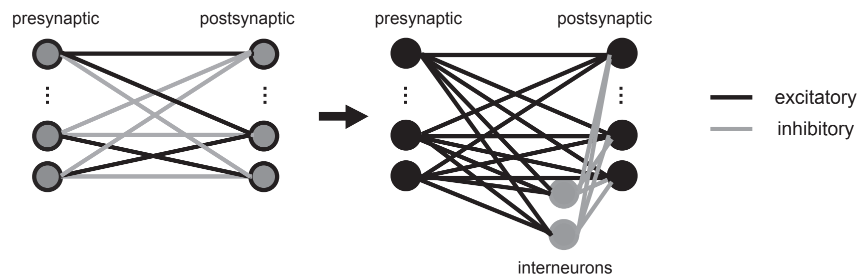

Figure 4: Conversion of an idealized projection model to a physiologically realistic form. The left diagram depicts an idealized projection from one population of neurons to another. Each of the pre-synaptic neurons is a source of both excitatory (black) and inhibitory (gray) synapses. Any idealized model of this form can be mapped onto the physiologically-realistic form on the right, which is typical of cortical anatomy. In this form, each pre-synaptic neuron is purely excitatory, and projects both directly to the post-synaptic neurons, and also indirectly through a smaller group of interneurons. See text for further details (section 3).

mixed signs, so that they are all excitatory. This introduces a bias current into the post-synaptic neurons, which is a function of the variable encoded by the pre-synaptic neurons. The pre-synaptic neurons also project to a small group of inhibitory interneurons, which project in turn to the post-synaptic neurons, compensating for the bias current that was introduced by increasing the excitatory weights. In this way, the original mapping is approximately recovered (Parisien et al., 2008). Because the indirect component of each projection passes through interneurons, it lags slightly behind the direct component. The idealized mapping is therefore slightly corrupted, proportionately with the derivative of the bias function. However, the time constant of synaptic current decay at excitatory synapses onto interneurons is often very rapid, on the order of 1ms (Geiger, Ĺubke, Roth, Frotscher, & Jonas, 1997; Walker, Lawrence, & McBain, 2002). The models are therefore parameterized with time constants of 1ms, which minimizes the lag-related corruption.

Results of example simulations with noisy ramp input are shown in Figure 5. In these simulations, the input signal (Figure 5A) passes through post-synaptic current dynamics, and is then injected as current into the neurons of the input ensemble of each network. The output of the circuit is an estimate of the variable encoded by the output ensemble, which is decoded from the activity of the neurons in this ensemble (see Appendix). Outputs of the idealized abstract models are overlaid with decoded outputs of the corresponding detailed spiking models. The output is noisy for each circuit, but much less noisy than the output of a pure differentiator, for which output noise is much larger than the range of the figure axes. Note that despite the nonlinearity of the depression model, its performance is comparable to that of the adaptation model, because other sources of error dominate.

Two main points can be taken from these simulation results. The first is that the response of each of the abstract models (dark lines) is slightly different. For example, there are differences in how quickly the different abstract models respond to the ramp, and in the amplitudes of their output noise. These differences are best described in terms of the abstract circuits' frequency responses, which are explored in detail in the next section. The second point is that the behavior of each circuit approximates the behavior of the corresponding abstract model, but it is noisier. Following the analysis of ideal frequency responses, we model this noise explicitly, and compare the noise that arises in each circuit.

4 Ideal Frequency Responses

In this section, we analyse the idealized frequency response characteristics of the example networks, in order to determine how they respond to input noise. We ignore for the moment that the networks are not only conduits of noise, but also sources, and return to this matter in the next section.

Laplace-domain transfer functions, obtained from the block diagrams (Figure 2) are listed in Table 1. These transfer functions describe the input-output behavior of the abstract models. Each transfer function has two or more poles, which reject high frequencies, and a single zero at the origin of the Laplace plane, which performs differentiation. (A zero is a

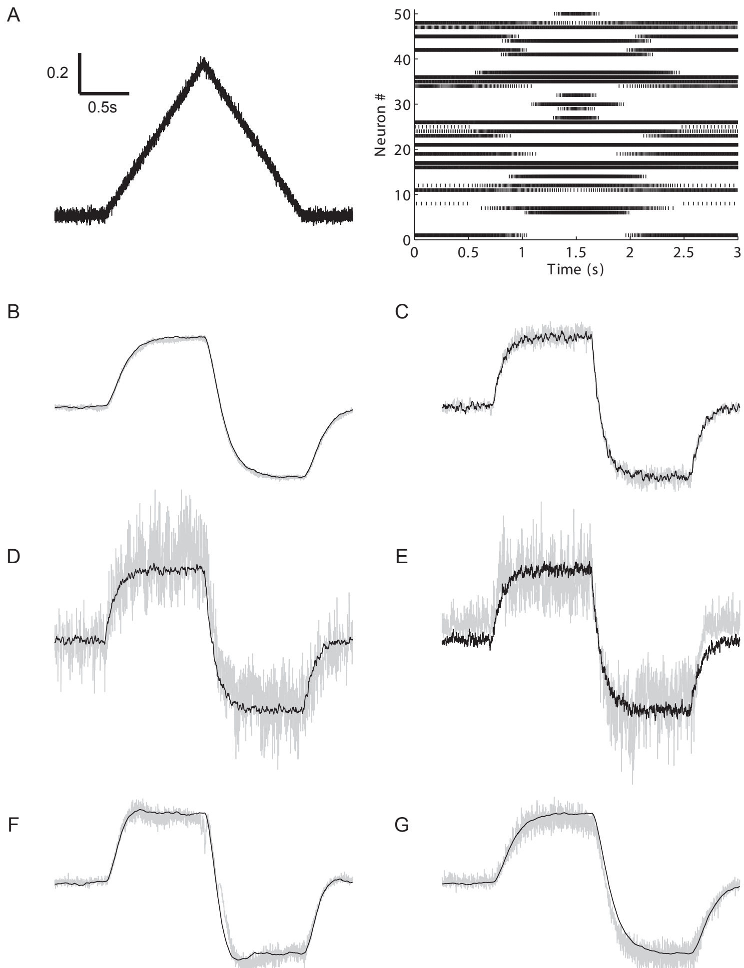

Figure 5: Example simulations with spiking LIF neurons. A, Noisy ramp input used in all simulations, and a spike raster showing spiking patterns of 50 neurons in the input ensemble (each row shows spike times for a single neuron; as the represented value increases, the firing rate of each neuron either increases or decreases, depending on its preferred direction $\tilde{\phi}_i$; see Appendix). The remaining panels show output of the different networks, for the input shown in panel A. Black lines indicate the output of abstract models, and gray lines indicate the decoded output of the corresponding neural network models. B, Feedforward network with an intermediate differentiator ensemble. C, Feedforward network in which parallel projections have different time constants. D, Network with adapting neurons. E, Network with synapses that exhibit short-term depression. F, Feedback network that approximates a Butterworth filter (scale factors tuned for this input; $p = [\ .1741 \quad .0774\ ]^T$). G, Feedback network with the same transfer function as the feedforward network of panel B ($p = [\ 1 \quad \frac{2}{3}\ ]^T$). The scale bars in A (0.5s; 0.2 normalized units of the represented variable) apply to all panels.

Table 1: Idealized transfer functions of the example differentiator networks (i.e. network output y over input u). These are given in terms of the Laplace variable s, and other parameters that describe the network dynamics. τ_A and τ_D are time constants of adaptation and synaptic depression, respectively. Other τ are time constants of post-synaptic current decay at various synapses: τ is the time constant of core differentiator synapses, which is modeled on NMDA receptors; τ_F corresponds to the faster parallel synapses in the dual-time-constant network (modeled on AMPA); τ_E corresponds to the faster synapses on the periphery of some networks (also modeled on AMPA). Finally, ω is the corner frequency of the Butterworth network. The values of these parameters for the analysis and simulations below are $\tau = \tau_A = \tau_D = 0.1s$, $\tau_F = 0.005s$, and $\omega = 2Hz$, except where these parameters are varied in section 6.

Circuit	Transfer Function
1.	$y/u = \frac{s}{(\tau s+1)^2}$
2.	$y/u = \frac{s}{(\tau_F s+1)(\tau s+1)}$
3.	$y/u = \frac{s}{(\tau_E s+1)^2(\tau_A s+1)}$
4.	$y/u = \frac{s}{(\tau_E s+1)(\tau_D s+1)}$
5.	$y/u = \frac{\omega^2 s}{(\tau_E s+1)(s^2+\sqrt{2}\omega s+\omega^2)}$
6.	$y/u = \frac{s}{(\tau s+1)^2(\tau_E s+1)}$

value of s for which the numerator of the transfer function vanishes, and a pole is a value at which the denominator vanishes; poles determine the time constants and resonance properties of a system.) The key differences are the pole locations. The poles of the circuits based on feedforward network dynamics are determined by the time constants of post-synaptic current decay. Similarly, those of the networks based on neuron-intrinsic dynamics are determined by both post-synaptic current dynamics and dynamics of adaptation/depression. In contrast, the poles of the networks based on feedback dynamics are not dictated by cell properties, but depend on the relationship between post-synaptic current dynamics and the strength of recurrent connections.

Phase and magnitude plots of the ideal and simulated frequency response of each circuit are shown in Figure 6. Intrinsic sources of noise were minimized in these simulations, by replacing each of the spiking neuron models used in the previous section with the corresponding rate model. The ideal responses (solid lines) and simulated responses (open circles) are similar, showing that the behavior of the optimized network models closely approaches that of the abstract dynamical models.

5 Intrinsic Error Sources

The key difference between the performance of the abstract circuits and the performance of the full spiking models (Figure 5) is that the spiking models are corrupted by much greater errors. To model the errors that arise within each circuit, we supplement the block diagrams with additive error terms ϵ_i, as shown in Figure 7. Each ϵ_i represents all the sources of error that arise within a single axonal projection. We consider two types of error. Part of the total error, which we call *distortion*, following engineering terminology, is a deterministic function of the variable that is encoded by a neuronal population. Specifically, distortion is the difference between the encoded value and the estimate of this value that is decoded from the activities of the neurons (post-synaptic neurons have access only to the latter). In contrast, we use the term *noise* to indicate error that is a random function of time. This includes noise arising during axonal conduction (Lass & Abeles, 1975), vesicle release (Stevens & Wang, 1994), etc. On the block diagrams, each ϵ_i term is shown as a function of both time, and of the variable encoded by the associated population, to indicate both the noise and distortion components. However, it should be noted that the errors are not strictly functions of these represented variables, but rather of dimensionless variables that are normalized to the representational range of each ensemble. Revised equations for the output of each system, as a function of both input and errors, are given in Table 2.

Some insights into the relative importance of the different error terms can be gained by direct inspection of the equations of Table 2. For example, in the first feedback circuit (#5 in Table 2) the differentiated components of ϵ_3 and ϵ_4 are amplified differently. With the

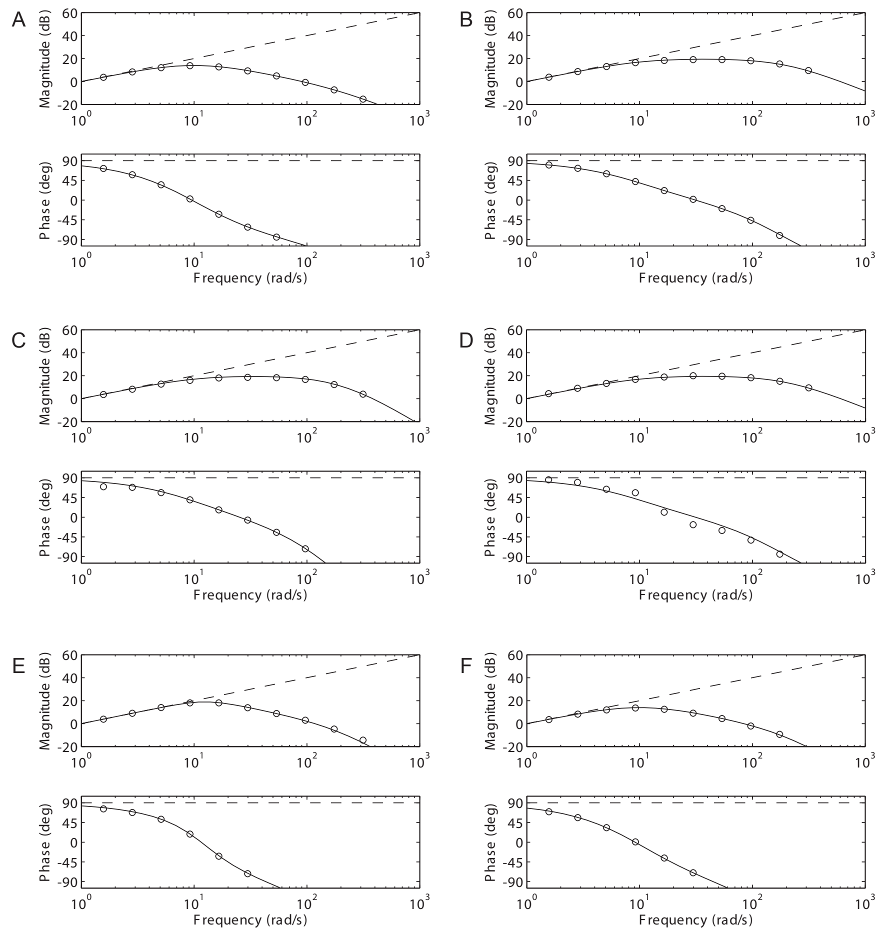

Figure 6: Magnitude and phase of the analytically-determined and simulated frequency responses of each network. A-F correspond to the block diagrams A-F in figure 2. The straight dashed line in each case corresponds to an ideal differentiator response. The solid black lines indicate theoretical responses of each network, and circles indicate results of simulations. The simulation results were obtained by running the networks in firing-rate mode, with sinusoidal input at various frequencies. Scale factors in the feedback networks are $p_1 = p_2 = 0.1741$ for E, and $p_1 = p_2 = 1$ for F.

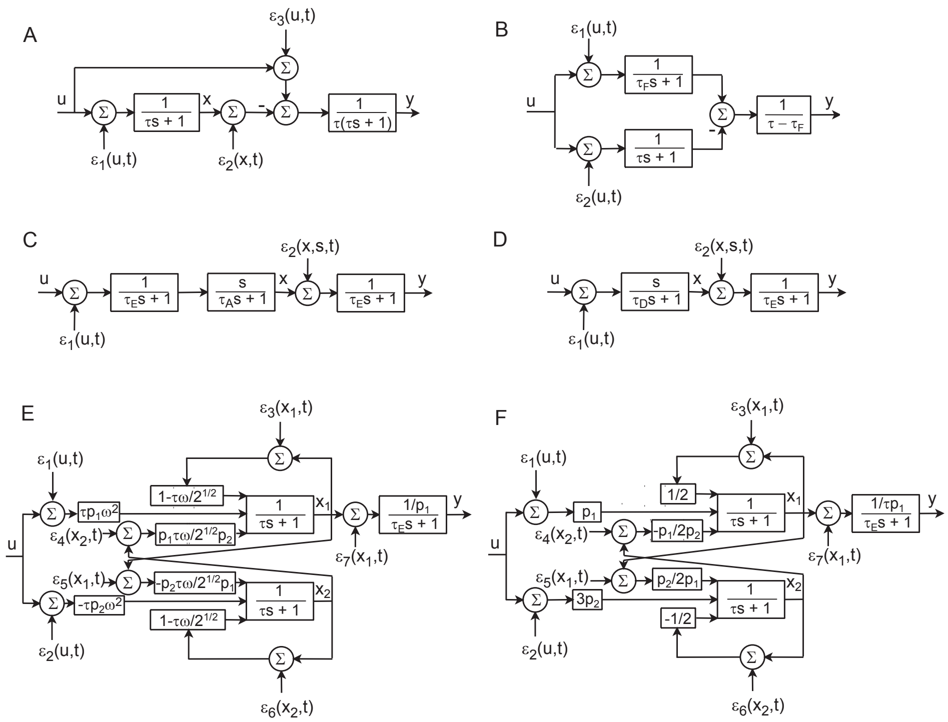

Figure 7: Revised block diagrams with additive error inputs ϵ_i. These error terms model distortion and noise that arise within each projection. A-F correspond to A-F in Figure 2.

Table 2: Output of differentiator circuits, including error terms. These equations are given in terms of the output y, rather than the transfer function y/u, in order to simplify inclusion of the errors ϵ_i. Other variables are as in Table 1.

Circuit	Circuit Output
1.	$y = \frac{su}{(\tau s+1)^2} - \frac{\epsilon_1}{\tau(\tau s+1)^2} + \frac{\epsilon_3-\epsilon_2}{\tau(\tau s+1)}$
2.	$y = \frac{su}{(\tau_F s+1)(\tau s+1)} + \frac{1}{\tau-\tau_F}\left[\frac{\tau s\epsilon_1-\tau_F s\epsilon_2+\epsilon_1-\epsilon_2}{(\tau_F s+1)(\tau s+1)}\right]$
3.	$y = \frac{su}{(\tau_E s+1)^2(\tau_A s+1)} + \frac{s\epsilon_1}{(\tau_E s+1)^2(\tau_A s+1)} + \frac{\epsilon_2}{(\tau_E s+1)}$
4.	$y = \frac{su}{(\tau_E s+1)(\tau_D s+1)} + \frac{s\epsilon_1}{(\tau_E s+1)(\tau_D s+1)} + \frac{\epsilon_2}{(\tau_E s+1)}$
5.	$y = \frac{\omega^2 su}{(\tau_E s+1)(s^2+\sqrt{2}\omega s+\omega^2)} + \frac{s(\omega^2\epsilon_1+(1-\tau\omega/\sqrt{2})\epsilon_3/\tau p_1+\omega\epsilon_4/\sqrt{2}p_2)}{(\tau_E s+1)(s^2+\sqrt{2}\omega s+\omega^2)} + \frac{\omega^3(\epsilon_1-\epsilon_2)/\sqrt{2}-\omega(\epsilon_3/p_1+\epsilon_6/p_2)(\tau\omega-\sqrt{2})/2\tau+\omega^2(\epsilon_4/p_2-\epsilon_5/p_1)/2}{(\tau_E s+1)(s^2+\sqrt{2}\omega s+\omega^2)} + \frac{\epsilon_7/p_1}{\tau_E s+1}$
6.	$y = \frac{su}{(\tau s+1)^2(\tau_E s+1)} + \frac{s(\epsilon_1+\epsilon_3/2p_1-\epsilon_4/2p_2)}{(\tau s+1)^2(\tau_E s+1)} + \frac{6(\epsilon_1-\epsilon_2)+3(\epsilon_3/p_1-\epsilon_4/p_2)+(\epsilon_6/p_2-\epsilon_5/p_1)}{4\tau(\tau s+1)^2(\tau_E s+1)} + \frac{\epsilon_7/\tau p_1}{\tau_E s+1}$

parameters used in the present study, ϵ_4 is amplified almost an order of magnitude more than ϵ_3. Since ϵ_4 arises from the second differentiator ensemble, this suggests that more of the available resources (i.e. neurons) should be devoted to accurately coding the corresponding state variable x_2 than x_1.

This type of isolated observation is interesting, but it does not allow systematic comparison of errors across different circuits. However, by making some additional assumptions, the distortion and noise in each circuit can be expressed as single quantities that correspond to their power at the output. First, we assume for the purpose of this analysis that each neuron is an independent source of Gaussian noise, with frequencies up to 500Hz. Then when a greater number n of neurons is used to represent a variable, noise power in the representation decreases with $1/n$, as predicted by the central limit theorem. Distortion error is a function of the value that is represented by an ensemble at any given time. The frequency content therefore depends on the frequency content of the input, although higher frequencies will be introduced by distortion non-linearities. We assume for this analysis that the error includes frequencies up to 200Hz. With linear decoding of diverse LIF neuron responses, mean-squared distortion error varies with population size as $1/n^2$ (Eliasmith & Anderson, 2003). Errors that arise within projections from different populations are assumed to be independent.

On the basis of these assumptions, net error power at the output can be calculated (using Parseval's theorem) as

$$E = \sum_i f(n_i) \int_{-\infty}^{\infty} |a_{i\omega} b_{i\omega}|^2 d\omega,$$

where n_i is the number of neurons in the population from which the i^{th} error term arises, $f(n_i)$ scales the error with population size (this function describes the expected magnitude of the error, which will be well-approximated for large n_i), $a_{i\omega}$ is the Fourier coefficient of the i^{th} raw error term at frequency ω, and $b_{i\omega}$ is the Fourier coefficient of the impulse response that corresponds to the transfer function through which this raw error term passes before reaching the output of the network. For example, using the assumptions outlined above, the total distortion error at the output of the intermediate-ensemble network (Figure 7A) is,

$$E_d = \frac{\epsilon_{d1} n_u^{-2}}{400} \int_{-200}^{200} \left| \mathcal{F}_\omega \left\{ \frac{t}{\tau^3} e^{-t/\tau} \right\} \right|^2 d\omega + \frac{\epsilon_{d1}(n_u^{-2} + n_x^{-2})}{400} \int_{-200}^{200} \left| \mathcal{F}_\omega \left\{ \frac{1}{\tau^2} e^{-t/\tau} \right\} \right|^2 d\omega.$$

Here $\mathcal{F}_\omega \{\bullet\}^2$ indicates power (of the error impulse response, i.e. the inverse Laplace transforms of the corresponding error transfer functions) at frequency ω, and ϵ_{d1} is the mean-squared distortion that would be expected from a single-neuron population (which is unknown, but assumed constant across networks). Because the distortion is assumed to have a white spectrum up to its cutoff frequency $\omega_c = 200Hz$, $a_{iw} = \frac{1}{\sqrt{2\omega_c}}$ for all ω within the integration limits $\pm\omega_c$.

This error expression allows more systematic inferences about the relative importance of different error sources. For example, it allows one to find the optimal distributions of neurons between differentiator ensembles in the feedback circuits. More importantly, this error expression makes it possible to directly compare the effects of intrinsic error between different networks. However, in order to quantify these errors for comparison across circuits, two additional subtleties must be accounted for. The first is that the sources of error in the adapting and depressing projections are not directly analogous to the others, so they cannot be compared directly. Rather than avoid the comparison completely, we present ranges of values based on the assumption that error arising from an adapting or depressing projection varies with pre-synaptic population size in the same way as other errors, but has a larger absolute value. Even if the adaptation and depression processes were themselves completely noise-free, the error amplitude would probably be at least five times greater. This is because 1) independent errors arising from adapting and compensating sub-populations sum together, and 2) the output of this projection must the amplified (because e.g. the adapting neurons do not adapt completely), so that the error sum is also amplified. We therefore consider a range (see Figure 9, below) of output error corresponding to between 5 and 50 times greater error amplitude in these projections relative to static projections.

The secondly subtlety is that some of the circuits contain parallel projections from the same ensemble, raising the possibility that errors in these separate projections are correlated. For example, the dual-time-constant network contains parallel projections with different PSC time constants. In a network of this form, similar plasticity processes might regulate the synaptic weights in each projection, resulting in correlations between the corresponding synaptic weights. This in turn would result in correlated distortion and/or noise. To determine how great an effect such correlations could have, we re-evaluate the output error, assuming that instantaneous errors in different projections from the same ensemble are equal. Setting $\epsilon_1 = \epsilon_2 = \epsilon$ in the dual-time-constant network, the output becomes

$$y = \frac{su}{(\tau_1 s + 1)(\tau_2 s + 1)} + \frac{s\epsilon}{(\tau_1 s + 1)(\tau_2 s + 1)},$$

which has very low error from intrinsic sources. Figure 8 compares simulations with and without this modification. Correlated weights do not reduce the amplification of input noise, but intrinsic sources of error are greatly reduced.

The interneuron and feedback circuits also have parallel projections from the same ensemble, which ultimately converge again. For example, the error transfer function of the interneuron circuit can be arranged to give the term $\epsilon_3 - \epsilon_1/(\tau s + 1)$, where ϵ_3 and ϵ_1 are potentially correlated. In this case, the phase shift between these errors depends on the frequency. At low frequencies the phase shift is zero, so these terms cancel out if $\epsilon_3 = \epsilon_1$.

Figure 9 shows the relative errors as a function of the size of the input population. For circuits 1, 5, and 6, which contain differentiator ensembles in addition to input and output ensembles, results are shown with different numbers of differentiator neurons. Panel A shows error power under the assumption that errors in parallel projections are independent, and panel B shows the lower error power at the output that arises from highly correlated errors.

The intermediate-ensemble network has lower error magnitude than the dual-time-constant network, because the fast time constant in the latter network does not filter high-frequency noise as sharply. The feedback and interneuron circuit errors do not fall as sharply as those of the other circuits, with increasing input ensemble size, because their errors arise partly from

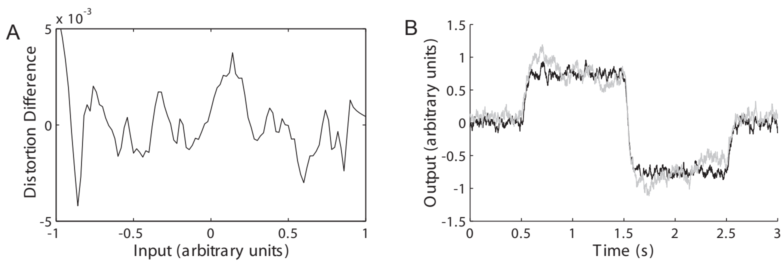

Figure 8: The performance of the two-time-constant feedforward network is improved if errors in each projection are correlated. A, Difference in distortion error between two independent parallel projections from a 2000-neuron ensemble, i.e. the distortion error (see section 5) of one projection minus that of another (independent distortion errors were modeled using optimal weights with independent ensembles). B, Output of network simulations with spiking LIF neurons and ramp input. Simulations with correlated (black line) and uncorrelated (grey line) distortion error are shown. To make the distortion more visible in these simulations, $\tau_2 = 0.015$s, and the spike output is filtered with a time constant of 20ms to reduce high-frequency noise. In this example, the root-mean-squared error of the correlated model output (.0616 arbitrary units) is comparable to that of an ideal filter with the same input noise (.0603; not shown), and substantially lower than that of the uncorrelated model (.1412).

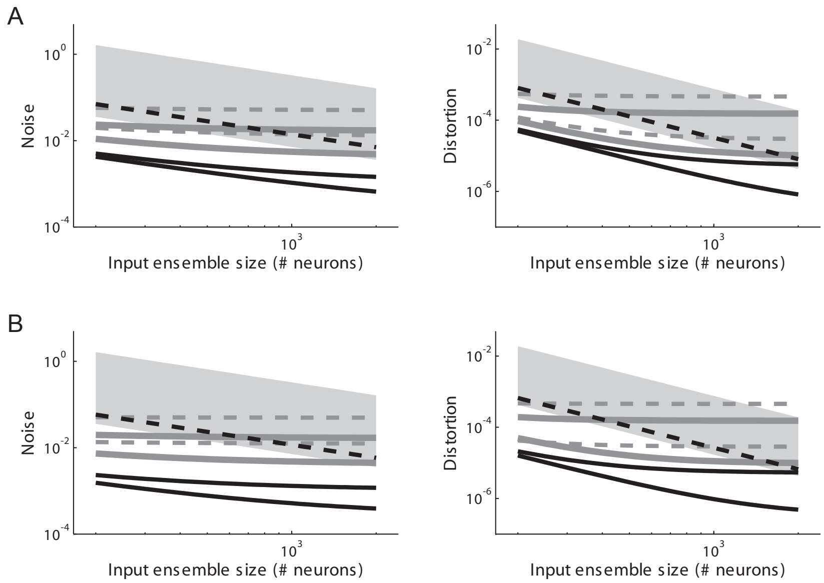

Figure 9: Power at the network outputs that arises from noise (left panels) and distortion (right panels) errors. Both noise and distortion have been estimated analytically using the methods and assumptions of section 5. Error in each panel is a dimensionless quantity which is relative to the error power expected in an ensemble of one neuron. For example, a projection from an ensemble of 200 neurons produces noise with relative power 1/200. A, Output error power as a function of input ensemble size, assuming that errors in parallel projections from the same ensemble are independent. Black lines show errors for the feedforward networks (solid: intermediate-ensemble network; dashed: dual-time-constant network). Gray lines show errors for the feedback networks (solid: Butterworth; dashed: intermediate-ensemble-like; scale factors as in Figure 6). Where there are two lines with the same pattern, these correspond to errors with 500 and 2000 differentiator neurons (higher and lower lines, respectively). Finally, the gray areas indicate the range of possible error for the adaptation network (with 2000 differentiator neurons), as described in the text (error in the depression network is similar). B, Identical to panel A except that errors in parallel projections from the same ensemble are assumed here to be highly correlated.

non-input neurons. The performance of the feedback circuits is enhanced due to additional low-pass filtering of errors by the post-synaptic dynamics of the output ensemble. In general, populations post-synaptic to the derivative calculation help to filter noise, and the noise they introduce has little effect on the output, because it is not differentiated. This implies that any of the circuits could be improved by passing the output through one or more additional ensembles with low-pass PSC dynamics, provided it is not necessary to differentiate high-frequency signals.

6 Dependence of Error on Time Constants

Error power at the output of each network depends on the time constants of post-synaptic current decay, adaptation, and depression. In the previous sections, the post-synaptic time constants in the main projections were modeled on the relatively slow NMDA glutamate receptors ($\tau = 0.1s$). This section shows how error magnitudes change as these time constants are reduced to the range of faster AMPA receptors ($\tau = .005s$). We also vary the time constants of adaptation and depression, which can be several seconds long in the brain.

Figure 10A shows how noise power depends on these time constants. Generally, slower dynamics attenuate high-frequency noise more effectively. However, signal and noise bands typically overlap, so that there is an inherent compromise between noise attenuation and signal distortion. Figure 10B shows how a signal with bandwidth 0-5Hz is also attenuated as these time constants increase. Figure 10C shows the corresponding signal-to-noise ratios.

In the feedforward and intrinsic-dynamics networks, a good balance of signal fidelity and noise attenuation is achieved when the time constants are well-matched to the signal band. The feedback circuits are distinct from the others, in that the signal power does not vary with the post-synaptic current time constant τ. Instead, it varies with the network time constants (e.g. ω), which depend on feedback strength. As a consequence, in a feedback circuit composed of neurons with given intrinsic dynamics, changes in synaptic weights can optimize the signal-to-noise ratio to the frequency band of the input signals. Longer intrinsic

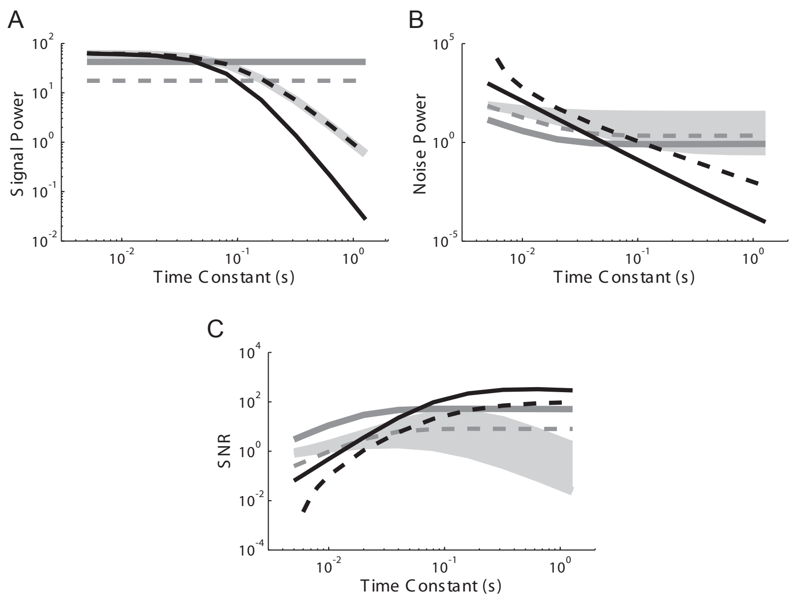

Figure 10: Signal and noise power as a function of neuron time constants (analytical results; arbitrary units). The same line patterns are used for each network as in Figure 9. A, Signal power: output power of each network (input signal power 1, bandwidth 0-5Hz; no noise), as a function of τ in the feedforward and feedback networks, and of τ_A and τ_D in the habituating-neuron networks (τ_F and τ_E are held constant at .005s). B, Noise power: output power of each network with zero input and independent noise (power 0.1, bandwidth 0-500Hz) in each projection. Distortion results (not shown) are similar. C, Signal-to-noise ratio: ratio of the power shown in panel A to that in panel B.

time constants reduce intrinsic noise, without influencing the signal.

7 Discussion

This study has analysed sources of error in networks that optimally estimate derivatives through diverse mechanisms. Interestingly, all of the examples studied were able to calculate a clear derivative signal. Therefore, in systems that contain a differentiator of unknown form, none of these structures can be entirely ruled out on the basis of performance. The main contrast between network types is that the feedforward and intrinsic-dynamic circuits have simple structures, and perform well for input signals with time scales that are matched to their internal dynamics, while the feedback circuits have more complex structures, and more flexible dynamic properties. This flexibility would be useful when the bandwidth of represented signals either does not correspond closely to membrane dynamics, or changes over time.

Part of the motivation for studying neural differentiators lies in the crucial role played by derivatives in feedback control. The network based on adaptation, in particular, may provide some insight into feedback control in the motor system. One way in which a derivative signal may be obtained in the motor system is through muscle spindle afferents. The group *Ia* afferents exhibit an adapting response to muscle stretch, while group *II* afferents respond statically. These responses are analogous to those in our adapting-neuron network, in which a pure derivative signal is extracted from a combination of adapting and non-adapting neuronal responses. Therefore, this network illustrates how distinct proportional and derivative signals can be extracted from muscle spindle afferents, allowing the brain to exploit these signals to improve feedback motor control.

There is a constraint on the flexibility of the feedback networks that is well worth noting, although it was not encountered in our examples. The eigenvalues of these networks' dynamics depend on the strength of the feedback connections. However, counter-intuitively, these circuits can become unstable with large negative eigenvalues. This is due to a hidden feedback mode that arises from the projection through inhibitory interneurons (Figure 4). This mode can become unstable with strong feedback of either sign. The precise stability threshold depends in complex ways on the neuronal tuning curves. However, in general, stability is reduced with stronger feedback, longer post-synaptic current time constants in the interneurons, and increasing slope of distortion error in the projection from the interneurons. This mode limits the range of stable eigenvalues in the feedback circuits, to between zero and roughly $-2/\tau$. Feedback circuits without an interneuron structure like that of Figure 4 would not have this limitation, which is only apparent in the more biologically realistic models. However, the additional phase lag that leads to instability could in principle be largely compensated by a phase lead arising from synaptic depression in this pathway.

The full spiking models performed more poorly than the abstract models, due to distortion in signal representations, and the high frequencies introduced by spiking. This suggests that neurons are not particularly good at differentiating, which is counter-intuitive, because natural neural circuits are evidently very powerful computational devices. However, when the brain differentiates, it must do so with the tools available, i.e. neurons, whether or not they are any good at that particular task. Differentiation is just one of many simple tasks (e.g. arithmetic; working memory) at which artificial systems apparently outperform biological systems.

7.1 Fractional Derivatives

The focus of this study has been on differentiation of order one. Higher-order derivatives could arise from a chain of any of the models presented here. Most of these models could also be modified to perform fractional differentiation (Kleinz & Osler, 2000). In the Laplace domain, a fractional derivative is expressed as s^k, where k is any real number. With $k < 1$, the fractional derivative of a signal is intermediate to the signal and its derivative. Repeated fractional derivatives may compose an integer-order derivative, e.g. $s^{\frac{1}{2}}s^{\frac{1}{2}} = s$. One reason fractional derivatives are interesting is that they are timescale-invariant, in that they phase-shift all frequencies by the same angle. This means, for example, that the fractional derivative of $x(t)$ has the same shape as the fractional derivative of $x(2t)$, although the amplitude increases as the timescale shortens. Timescale-invariant dynamics (Drew & Abbott, 2006; Wark et al., 2007) contrast with exponential dynamics, which have a distinct timescale. Firing-rate adaptation in some pyramidal cells is more closely approximated by a fractional differentiator than a band-pass filter, when responses are considered over several seconds (Lundstrom et al., 2008), in that firing rates do not asymptotically approach positive values after many multiples of the time constant of their initial decay. Fractional behavior has been proposed to arise from diverse timescales of intrinsic adaptive processes (Lundstrom et al., 2008).

A fractional derivative can be approximated as a weighted sum of high-pass responses with different time constants (Anastasio, 1994). Introducing multiple time constants into the above models should therefore (with proper weighting) produce fractional-order behavior. Introducing multiple time constants into the adapting-neuron network is straightforward. Recall that in this model, extra steps were taken to ensure that all of the neurons adapted with the same time constant. Variations across a population in the slope of the onset response function, and/or adaptation parameters, could lead to diverse time constants and fractional dynamics at the population level. A model of such a system could be constructed by using the methods of section 2.2.2 to optimize the distribution of time constants for a fractional response. This mechanism would be independent of inherently-fractional behavior within adapting neurons, and might serve to extend the frequency range of fractional behavior, or to compensate for deviations from ideal fractional behaviour due to cell-intrinsic mechanisms alone. Interestingly, if the firing rates of different neurons were to adapt with different time constants, multiple alternative transfer functions might be supported by different sets of synaptic weights applied to the same population activity.

The other mechanism that can provide a wide range of effective time constants in these models is feedback. For example, the feedback networks described above could be divided into multiple parts with different time constants. If these sub-networks were segregated, then fractional behavior would arise at the population level (hence in downstream neurons), while individual neurons in the sub-networks exhibit integer-order responses. More generally, individual neurons might participate in distributed coding of multiple state variables with different dynamics, and therefore exhibit behaviour at multiple time scales.

In sum, fractional differentiation may serve a number of purposes in the brain, e.g. compensating for fractional plant dynamics (Anastasio, 1994), or introducing other effects of multiple-timescale behavior such as optimization of information transmission (Fairhall, Lewen, Bialek, & Ruyter Van Steveninck, 2001). Experimental evidence suggests that cell-intrinsic mechanisms occur on multiple timescales, contributing to cell responses that resemble fractional derivatives. Population and network mechanisms like those explored in this paper might improve this approximation, and/or extend this behavior over wider frequency ranges. The present analytical approach could be used in the future to quantify the limitations of fractional network mechanisms in detail. For example, while positive feedback can theoretically produce a very wide range of time constants, attractors arise from long time constants combined with distortion in the feedback (Eliasmith & Anderson, 2003). This may suggest that relatively more neurons would be needed to code slower timescales, or that slower timescales are managed more effectively by cell-intrinsic mechanisms.

7.2 Integrator Control

A suitable differentiator might reduce drift in a neural integrator. If an input signal is integrated over time, then the derivative of the integrator output could be compared to the input signal. The resulting error information could then, in theory, be used as a proportional control signal to improve an imperfect integrator. However, since the differentiator is also imperfect, it is not guaranteed that this additional complexity would improve performance. Analysis of this feedback loop shows that improvement is possible if errors in parallel differentiator projections are correlated (not shown). The adapting network might also improve performance of an integrator, if the adapting projection errors were not much greater than the others.

Perhaps more interestingly, a slight variation of the adapting-neuron differentiator might directly implement an accurate integrator. It is well known that integrator error decreases with longer post-synaptic current (PSC) time constants (Seung, 1996). However, even the relatively long PSC time constants associated with NMDA and $\mathrm{GABA_B}$ receptors are only about 0.1s. In our adapting-neuron example, non-adapting neurons cancelled out the steady-state response of the ensemble, resulting in a pure differentiator. If instead these non-adapting neurons were to cancel out the onset response, the ensemble would behave like a low-pass filter. This low-pass filter would have the time constant of adaptation, which could be much slower than NMDA and $\mathrm{GABA_B}$ currents, e.g. on the order of seconds (Sanchez-Vives, Nowak, & McCormick, 2000b, 2000a). However, populations in such a circuit would have to encode variables with larger values (if u is the input and x is the integral, they would have to encode $x + \tau_A u$). Distortion error scales with the magnitude of values to be encoded, so that increasing τ_A indefinitely would not yield any benefit. However, there might be a range of longer adaptation time constants with better performance, particularly if inputs were small relative to their integrals. Detailed exploration of the relation between these differentiator circuits and neural integration is deferred for future work.

7.3 Limitations

While the differentiator circuits presented here have a degree of physiological realism, they are idealized in several respects. For example, we have modelled the networks with least-squares optimal synaptic weights, without addressing how these weights could be learned. We also used simple point-neuron models in the simulations, which neglect many details of the dynamics of real neurons, e.g. low-pass effects of spiking nonlinearities in some regimes (Fourcaud-Trocmé, Hansel, Vreeswijk, & Brunel, 2003). The present study would have been intractable without these simplifications.

Secondly, the example networks presented here do not exhaust all the possibilities. For instance, we have also modeled two varieties of differentiator that are based on delay lines (not shown). The first has two parallel feedforward projections, of which one has a longer axonal conductance time. In the second, conductance times in a feedforward projection vary from zero to thirty milliseconds, and synaptic weights are chosen so that the projection acts as a finite-impulse-response filter. These networks are comparable to the other feedforward networks presented here. However, there may be other differentiator models that do not belong to any of the classes we have studied. For example, a model that discriminates between low and high velocities, using a dendritic spike-related mechanism (Tamosiunaite, Porr, & Wörgötter, 2007) might be extended to encode velocity signals in more detail.

Furthermore, the analysis was based on a number of assumptions that may not hold for biological networks. For example, there is growing evidence that noise does not scale with population size in the manner we have assumed, because response variability in different neurons is correlated (although whether it is reasonable to consider such correlated variability as noise is unclear). Also, while our network models optimally approximate ideal filters, biological networks may approximate ideal differentiators in a non-optimal manner. Both of these assumptions affect estimates of error amplitudes. However, the present analytical approach easily accommodates changes to these assumptions, if such changes seem warranted in the future.

Some models of neural integrators have been criticized because they require precise tuning of recurrent excitation, to avoid drift (Goldman et al., 2003; Koulakov et al., 2002), and it is not clear how well such precise tuning can be maintained. Analogously, the differentiator circuits presented here require a precise balance of excitation and inhibition. The consequence of imbalance is increased distortion error, the effects of which are described in Section 5. A related question is whether there are any signals available to contribute to fine-tuning this balance. Over long periods, the average derivatives of many physiologically-relevant signals, such as the outputs of sensory and proprioceptive receptors, must be zero. We therefore suggest that if the output signal of a neural differentiator could be accurately averaged or filtered over a long time window, this might provide a useful error signal for fine-tuning the circuit.

Finally, further work is needed to estimate the degree to which the nonlinearity of synaptic-depression (section 2.2.2) is functionally relevant, particularly as compared with the more subtle nonlinearities that would appear in more detailed models of adaptation. This depends mainly on the magnitude of this nonlinearity, relative to other sources of error. In simulations presented here, other sources of error were dominant, so the performance of the synaptic depression model approached that of the adaptation model. Differences between these models might become more obvious if other sources of error were reduced, e.g. with greater numbers of neurons, but it is uncertain how great a reduction could be achieved with physiologically realistic parameters. Also, it might be possible to modify the model so that depression nonlinearities in different neurons cancelled out as much as possible at the network level (much like the static nonlinearities in the present models). Finally, the effect of a nonlinearity depends (by definition) on the input signal. In order to quantify the effects of the depression nonlinearity in detail, it would be necessary to examine responses to a much wider variety of inputs. The present results show that for certain inputs, in the context of certain plausible levels of noise and other network parameters, synaptic depression is functionally similar to firing-rate adaptation.

7.4 Summary

We have analyzed and contrasted three general mechanisms by which neural circuits can perform temporal differentiation of input signals. The analysis and accompanying simulations estimate in detail how the key barriers to accurate differentiation, i.e. representational distortion and high-frequency noise, depend on intrinsic network properties. The results suggest that no single mechanism is clearly superior, but that different mechanisms can be expected to perform most effectively with input signals in different frequency bands. The analytical approach developed here provides a systematic means of contrasting the performance of diverse networks, and may therefore provide a useful foundation for more detailed models.

Finally, although we have focused on differentiators, the techniques presented for analysing the effects of noise and distortion are more broadly applicable. It should be possible to analyse networks that perform other computations in a similar manner. It may also be possible to rule out certain mechanisms, or even to rule out a proposed function for a certain circuit, on the basis of this type of analysis. Finally, if the function of a circuit is known, the same methods could be used to determine bounds on the errors that can be introduced by different ensembles, possibly shedding light on the robustness of a circuit to degeneration with disease.

8 Acknowledgements

We are grateful to Charles H. Anderson, Chris Parisien, Abninder Litt, Brian N. Lundstrom and two additional anonymous reviewers for their helpful comments. This work is supported by the Canada Research Chairs Program, the National Science and Engineering Research Council of Canada (CGS-D and 261453-05), the Canadian Foundation for Innovation (3358401), and the Ontario Innovation Trust (3358501).

9 Appendix: Theoretical Framework

In the differentiator circuits presented here, firing activity in each neural ensemble encodes a scalar variable. We begin by describing the encoding process in the input ensemble, which (identically in each circuit) receives external input in the form of a scalar variable u. The activity of each neuron in this ensemble can be expressed in terms of a firing rate $r(u)$ which is a function of net current $I(u)$ flowing into the neuron's axon hillock. We use leaky-integrate-and-fire (LIF) neurons (Koch, 1999), for which

$$r(u) = \frac{1}{\tau_{ref} - \tau_{RC} \ln(1 - I_{th}/I(u))},$$

where τ_{ref} is the spike refractory time, τ_{RC} is the membrane time constant, and I_{th} is the current at which the firing rate becomes non-zero. In our models, $\tau_{RC} = .02s$, τ_{ref} varies between $.0005s$ and $.001s$, and $I_{th} = 1$.

In non-adapting neuron models, the net current is

$$I(u) = \alpha\tilde{\phi}u + \beta,$$

where α is a scaling factor, $\tilde{\phi}$ is the neuron's preferred direction in the encoded space (-1 or 1 since we are dealing exclusively with scalars) and β is a bias current. The bias current models intrinsic conductance and input bias. For adapting LIF neurons (Camera et al., 2004),

$$I(u) = \alpha\tilde{\phi}u + \beta - g_N[\mathrm{N}]$$

$$d[\mathrm{N}]/dt = -[\mathrm{N}]/\tau_N + A_N r(u).$$

Here [N] is the concentration of a chemical species responsible for adaptation, τ_N is a time constant of decay of [N], A_N is an increment in [N] with each spike, and g_N scales the associated current.

The firing rates of neurons in the input ensemble contain information about the input u, and an estimate of u can be decoded through a linear combination of the activities of each neuron, as

$$\hat{u} = \sum_i \phi_i r_i(u).$$

Here $\hat{u}$ is the estimate of u from the neuronal activities, and ϕ_i are decoding coefficients. Unconstrained optimal decoding coefficients are found by minimizing the squared error of the estimate using the Moore-Penrose pseudo-inverse. This decoding of neuronal activity allows us to plot the scalar that is represented by an ensemble over time (e.g. see Figure 5).

In some cases, neurons are simulated in spiking mode. In this case, the activity $a(u)$ of a neuron is not expressed as a scalar-valued rate $a(u) = r(u)$, but a series of spike events

$a(u) = \sum_k \delta(t_k - t)$, where t_k is the time at which the neuron spikes for the k^{th} time. By treating each spike as a unit impulse, and passing spikes through low-pass dynamics (of the same form as post-synaptic current dynamics), we obtain a similar estimate of u.

The encoding and decoding processes described above allow us to find the synaptic weights that are necessary for the input ensemble to communicate its estimate of u to another ensemble. Suppose $x \simeq u$ is the scalar variable represented by a second ensemble. The current in these neurons is a function of $\hat{u}$ rather than u, e.g.

$$I_j(u) = \alpha\tilde{\phi}_j\hat{u} + \beta = \alpha\tilde{\phi}_j \sum_i \phi_i r_i(u) + \beta = \sum_i w_{ij} r_i(u) + \beta.$$

Here $w_{ij} = \alpha\tilde{\phi}_j\phi_i$ is the synaptic weight (between the i^{th} neuron in the first ensemble and the j^{th} neuron in the second ensemble) that is necessary to communicate the estimate of u from one ensemble to the other. Similarly, the synaptic weights necessary to communicate a multiple of $\hat{u}$, e.g. $c\hat{u}$ from one ensemble to another are $w_{ij}^{\beta} = \alpha\tilde{\phi}c\phi$. This expression allows us to construct neuronal circuits in which external inputs are encoded, communicated, transformed, and decoded as required. For simulation, these optimal synaptic weights are converted to corresponding sign-constrained weights, as described in Section 3.

For a simple circuit, it is not difficult to write equations for the activities of each neuron, in terms of the input signal. This process is not necessary for developing a computational model, but an example may clarify the computations involved. Consider the dual-time constant circuit (Figure 2B). For simplicity we ignore noise sources, and write the equations in terms of neuronal firing rates rather than spike events (the equations are the same for spiking simulations, except that real-valued rates are replaced with sums of impulses). The input to the circuit is a scalar function of time, u. Although u is an abstract signal, we treat it like a synaptic input and pass it through post-synaptic current dynamics with impulse response function $h(t) = \frac{1}{\tau}e^{-t/\tau}$, because physiological inputs to these neurons are filtered in this manner. The firing rate of the i^{th} neuron in the input ensemble is then

$$r_i(u) = G_i(\alpha_i \tilde{\phi}_i u(t) * (\frac{1}{\tau} e^{-t/\tau}) + \beta_i),$$

where $*$ denotes convolution, and $G_i(\cdot)$ is the rate function for the LIF neuron, which is given above. Similarly, the firing rate of the j^{th} neuron in the output ensemble is

$$r_j(u) = G_j(\alpha_j \tilde{\phi}_j (\sum_i \phi_i r_i(u)) * (\frac{1}{\tau_F} e^{-t/\tau_F} - \frac{1}{\tau} e^{-t/\tau}) + \beta_j),$$

where τ_F and τ are the fast and slow time constants of this circuit, as defined in the main text. Finally, the output of the circuit, which approximates du/dt, is taken as a linear decoding of the activity of the output ensemble:

$$\hat{y} = \sum_j \phi_j r_j(u).$$

The above description provides all the theoretical details that are necessary for implementing the models in the present study. The source code for the computational model implementations can be downloaded from http://compneuro.uwaterloo.ca. Further details of the theoretical methods, including generalization to the representation of vectors and functions, have been presented previously (Eliasmith & Anderson, 2003).

References

Abbott, L. F., & Regehr, W. G. (2004). Synaptic computation. *Nature*, *431*, 796-803.

Anastasio, T. J. (1994). The fractional-order dynamics of brainstem vestibulo-oculomotor neurons. *Biol Cybern*, *72*, 69-79.

Benda, J., & Herz, A. (2003). A universal model for spike-frequency adaptation. *Neural Computation*, *15*, 2523-2564.

Camera, G. L., Rauch, A., Lüscher, H. R., Senn, W., & Fusi, S. (2004). Minimal models of adapted neuronal response in in vivo-like input currents. *Neural Computation*, *16*, 2101-2124.

Drew, P. J., & Abbott, L. F. (2006). Models and properties of power-law adaptation in neural systems. *J Neurophysiol*, *96*, 826-33.

Dunn, N. A., Lockery, S. R., Pierce-Shimomura, J. T., & Conery, J. S. (2004). A neural network model of chemotaxis predicts functions of synaptic connections in the nematode caenorhabditis elegans. *J Comp Neuro*, *17*, 137-147.

Eliasmith, C. (2004). A unified approach to building and controlling spiking attractor networks. *Neural Comput*, *17*, 1276-1314.

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems.* Cambridge: MIT Press.

Fairhall, A. L., Lewen, G. D., Bialek, W., & Ruyter Van Steveninck, R. R. de. (2001). Efficiency and ambiguity in an adaptive neural code. *Nature*, *412*, 787-792.

Fourcaud-Trocmé, N., Hansel, D., Vreeswijk, C. van, & Brunel, N. (2003). How spike generation mechanisms determine the neuronal response to fluctuating inputs. *J Neurosci*, *23(37)*, 11628-11640.

Geiger, J. R. P., L´ubke, J., Roth, A., Frotscher, M., & Jonas, P. (1997). Submillisecond ampa receptor-mediated signaling at a principal neuron-interneuron synapse. *Neuron*, *18*, 1009-1023.

Goldman, M. S., Levine, J. H., Major, G., Tank, D. W., & Seung, H. S. (2003). Robust persistent neural activity in a model integrator with multiple hysteretic dendrites per neuron. *Cereb Cortex*, *13*(11), 1185-1195.

Goldman, M. S., Maldonado, P., & Abbott, L. F. (2002). Redundancy reduction and sustained firing with stochastic depressing synapses. *J Neurosci*, *22(2)*, 584-591.

Holstein, G. R., Rabbitt, R. D., Martinelli, G. P., Friedrich, V. L., Boyle, R. D., & Highstein, S. M. (2004). Convergence of excitatory and inhibitory hair cell transmitters shapes vestibular afferent responses. *Proc Nat Acad Sci*, *101*(44), 15766-15771.

Jenison, R. L. (2007). Neural dynamics in human auditory cortex. In *Cnc conference: Direct recordings from the human brain.*

Joel, D., Niv, Y., & Ruppin, E. (2002). Actor-critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Networks*, *15*, 535-547.

Kawada, T., Uemura, K., Kashihara, K., Kamiya, A., Sugimachi, M., & Sunagawa, K. (2004). A derivative-sigmoidal model reproduces operating point-dependent baroreflex neural arc transfer characteristics. *Am J Physiol Heart Circ Physiol*, *286*, H2272-H2279.

Kleinz, M., & Osler, T. J. (2000). A child's garden of fractional derivatives. *The College Mathematics Journal*, *31*, 82-88.

Koch, C. (1999). *Biophysics of computation: Information processing in single neurons.* New York, NY: Oxford University Press.

Koulakov, A., Raghavachari, S., Kepecs, A., & Lisman, J. (2002). Model for a robust neural integrator. *Nature Neuroscience*, *5*(8), 775-782.

Lass, Y., & Abeles, M. (1975). Transmission of information by the axon: I. noise and memory in the myelinated nerve fiber of the frog. *Biological Cybernetics*, *19*(2), 61-67.

Liu, Y. H., & Wang, X. J. (2001). Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron. *J Computational Neuroscience*, *10*, 25-45.

Lundstrom, B. N., Higgs, M. H., Spain, W. J., & Fairhall, A. L. (2008). Fractional differentiation by neocortical pyramidal neurons. *Nature Neuroscience*, *11*, 1335-42.

Miller, P., Brody, C. D., Romo, R., & Wang, X. J. (2003). A recurrent network model of somatosensory parametric working memory in the prefrontal cortex. *Cerebral Cortex*, *13*, 1208-1218.

Naud, R., Berger, T., Badel, L., Roth, A., & Gerstner, W. (2008). Quantitative single-neuron modeling: competition 2008. In *Frontiers in neuroinformatics.*

Parisien, C. M., Anderson, C. H., & Eliasmith, C. (2008). Solving the problem of negative synaptic weights in cortical models. *Neural Computation*, *20(6)*, 1473-94.

Poon, C. S., Young, D. L., & Siniaia, M. S. (2000). High-pass filtering of carotid-vagal influences on expiration in the rat: role of n-methyl-d-aspartate receptors. *Neuroscience Letters*, *284*, 5-8.

Puccini, G. D., Sanchez-Vives, M. V., & Compte, A. (2007). Integrated mechanisms of anticipation and rate-of-change computations in cortical circuits. *PLoS Computational Biology*, *3*, 813-25.

Robinson, D. A. (1968). A note on the oculomotor pathway. *Experimental Neurology*, *22*, 130–132.

Robinson, D. A. (1989). Integrating with neurons. *Annual Review of Neuroscience*, *12*, 33–45.

Sanchez-Vives, M. V., Nowak, L. G., & McCormick, D. A. (2000a). Cellular mechanisms of long-lasting adaptation in visual cortical neurons in vitro. *Journal of Neuroscience*, *20*(11), 4286-4299.

Sanchez-Vives, M. V., Nowak, L. G., & McCormick, D. A. (2000b). Membrane mechanisms underlying contrast adaptation in cat area 17 in vivo. *Journal of Neuroscience*, *20*(11), 4267-4285.

Sedra, A. S., & Smith, K. C. (1991). *Microelectronic circuits, 3 ed.* Saunders College Publishing.

Seung, H. S. (1996). How the brain keeps the eyes still. *Proc Nat Acad Sci*, *93*, 13339-13344.

Stevens, C. F., & Wang, Y. (1994). Changes in reliability of synaptic function as a mechanism for plasticity. *Nature*, *371*, 704-707.

Tamosiunaite, M., Porr, B., & Wörgötter, F. (2007). Developing velocity sensitivity in a model neuron by local synaptic plasticity. *Biological Cybernetics*, *96*, 507-518.

Topka, H., Mescheriakov, S., Boose, A., Kuntz, R., Hertrich, I., Seydel, L., et al. (1999). A cerebellar-like terminal and postural tremor induced in normal man by transcranial magnetic stimulation. *Brain*, *122*, 1551-1562.

Vegte, J. Van de. (1994). *Feedback control systems.* Englewood Cliffs, NJ: Prentice-Hall.

Walker, H. C., Lawrence, J. J., & McBain, C. J. (2002). Activation of kinetically distinct synaptic conductances on inhibitory interneurons by electrotonically overlapping afferents. *Neuron*, *35*, 161-171.

Wark, B., Lundstrom, B. N., & Fairhall, A. (2007). Sensory adaptation. *Curr Opin Neurobiol*, *17(4)*, 423-29.

Yoshida, K., Watanabe, D., Ishikane, H., Tachibana, M., Pastan, I., & Nakanishi, S. (2001). A key role of starburst amacrine cells in originating retinal direction selectivity and optokinetic eye movement. *Neuron*, *30*, 771-780.

Young, D. L., Eldridge, F. L., & Poon, C. S. (2003). Integration-differentiation and gating of carotid afferent traffic that shapes the respiratory pattern. *J Appl Physiol*, *94*, 1213-1229.

Zucker, R. S., & Regehr, W. G. (2002). Short-term synaptic plasticity. *Annu Rev Physiol*, *64*, 355-405.