

Non-Information-Maximizing Neural Coding

Michael Stiber

Computing & Software Systems

University of Washington Bothell, WA, 98011-8246 USA

Email: stiber@u.washington.edu

Abstract—Information theoretic techniques are often used to investigate neural coding. Results — in terms of bits per second or bits per spike — have been used as evidence to support temporal or rate coding, spike timing precision, etc. Despite its use this way, information theory does *not* tell one what the neural code (or any code) is. In artificial systems, codes are often purposefully made sub-optimal from a pure information density point of view. This work tests the feasibility of a neural code containing error correction characteristics which uses greater spike timing precision than might be necessary to simply transmit a given amount of information. A model of the recognized prototype of an inhibitory synapse shows that, even compared to small input imprecision and in the presence of robust dynamical behaviors, high timing precision can enable error correction.

Index Terms—spiking neurons, information theory, error correction, neural coding, nonlinear dynamics.

I. INTRODUCTION

INFORMATION theory provides general techniques for characterizing signal content or channel capacity. As such, it has been used with increasing regularity as a tool for investigating neural coding [1], [2], [3]. By comparing neuron output with sensory stimuli or presynaptic trains, researchers have concluded that certain preparations utilize temporal [4] or rate [5] coding, what their channel capacity is [6], etc.

What often has been lost in this work, however, is that information theory does *not* inform about the nature of the code used by a system, but only how efficient a carrier of information it is. In other words, it tells you how close a code is to being “optimally compressed”. Moreover, one can only learn about a channel’s characteristics if one knows what the channel *is*: for example, the distinction between the encoded data transmission and channel noise (source vs. channel coding). Because of this, conclusions about neural coding must be made carefully:

- Presynaptic spike trains are sometimes used as the reference in determining postsynaptic information capacity [3]. This treats a neuron as a pure encoder. However, neurons are dynamical systems, and their internal state certainly affects their discharge. To the extent that a neuron’s internal state represents information to be transmitted to other cells, these approaches underestimate capacity.
- The idea that neurons are noisy and unreliable has great currency in the literature, based on observations that spike trains look “messy” and that multiple experiments with identical stimuli produce different discharges. These are

weak definitions of noise: “variation of a measured quantity from some expected value” in the first case and “whatever is not of interest” in the second [7]. Use of spike train variability as a noise floor in channel capacity estimation only establishes a lower bound on information content, as it ignores the effects of any persistent internal state.

- An apparently low information rate (in bits per spike) might be taken as evidence for rate coding. This presupposes that one knows what data is being coded, arguably only feasible for sensory cells [1], [6], [8], [4], [5]. In any event, this at best establishes that the neural code is not optimally compressed, which is not the same thing as determining that there is no “extra content” in the code.
- In artificial systems, there are many reasons for not using optimally compact codes, including convenience, compatibility with other systems, encoding/decoding complexity, and error correction. The observation that a code contains redundancy is not the same as determining that the redundant information is not used. For example, one could analyze the information content of an audio compact disc: the fact that each bit of data carries less than one bit of information does not mean that not all of the bits are used (nor does it tell one what the code is or what is being encoded).
- It is not unreasonable that similar considerations may be applicable to biological systems. Neuron construction and neural architectures have definite (though unknown) limits to their processing power. If we assume that neuron operation contains significant “noise” uncorrelated with information to be transmitted, then this would be expected to inject errors into the data stream. This argues for codes with significant, and useful, redundancy.

The work described here is a preliminary test of the feasibility of the hypothesis that neural codes can contain redundancy for the purpose of error correction. It presents a simple, biologically plausible system that performs its function more reliably if provided with “extra information”. More specifically, a model of a prototypical inhibitory synapse and postsynaptic neuron is presented with low and high temporal precision pacemaker presynaptic (input) spike trains with and without errors. Error aftereffects are then compared to determine if higher temporal precision — more information in the input — can support faster recovery (faster return to the nominal output) — greater error-correction capability.

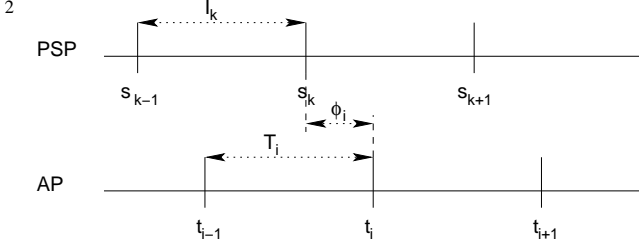


Fig. 1. Point process analysis. Analysis is based on intervals (I_k , T_i) and cross intervals (ϕ_i).

II. METHODS

A well-characterized model of the crayfish slowly-adapting stretch receptor organ (SAO) [9] was used. The living preparation includes the recognized prototype of an inhibitory synapse; the model, summarized in the Appendix, reproduces well the behaviors seen in the living preparation [10], [11]. Simulations were conducted using the ODEPACK library [12] in C, with input generation and analysis performed using MATLAB [13].

In the absence of inputs, the SAO acts as a pacemaker, producing action potentials (APs, outputs) all separated by its *natural* interval, N . Inputs are presented as postsynaptic potentials (PSPs) produced by transient changes in membrane chloride permeability. In this work, the PSP and AP trains are assimilated to point processes (i.e., only their times of occurrence are used). Figure 1 presents the essential terminology. PSPs and APs are numbered, with PSP k occurring at time s_k and AP i at t_i . From these times, presynaptic intervals I_k , postsynaptic intervals T_i , and cross intervals (or *phases*) ϕ_i can be computed.

When a nonlinear oscillator such as the SAO is presented with periodic input ($I_k = I, \forall k$), a number of stereotypical behaviors are possible. In both the SAO and this model, phase locking, quasiperiodicity, and chaos have been observed [14], [15], [16], [17], [18], [11]. For this preliminary work, PSP timing and strength was selected to produce 1:1 locking. In a $p:q$ locked behavior, p PSPs and q APs occur in the same period of time, $pI = qT$, where T is the average postsynaptic interval. Locking is periodic, so the sequence of intervals T_i and cross intervals ϕ_i repeat every q (i.e., $T_i = T_{i+q}$ and $\phi_i = \phi_{i+q}$). These q intervals are readily apparent as horizontal lines in *basic plots*: plots of T_i or ϕ_i versus t_i . The fact that the intervals repeat with period q can be verified by a *return map* plotting (T_i, T_{i+q}) or (ϕ_i, ϕ_{i+q}). In such a plot, all intervals in a $p:q$ locked behavior would fall on the diagonal ($T_i = T_{i+q}$ or $\phi_i = \phi_{i+q}, \forall i$).

To determine the perturbations introduced by errors, a *canonical*, high-temporal-precision PSP train, $\mathcal{H} = \{s_1, s_2, \dots, s_n\}$ was first generated (Figure 2(a), left). Pacemaker stimulation was used, so all PSPs were separated by an invariant interval, I (i.e., $I_k = I, \forall k$). The interval I was chosen so that the model would produce 1:1 locking. Errors were introduced by eliminating PSPs (b, dotted lines) [19]. The set of PSPs to be eliminated, $\mathcal{K} = \{k_1, k_2, \dots, k_m\}$, at times $\mathcal{S} = \{s_{k_1}, s_{k_2}, \dots, s_{k_m}\}$ (the *errors*) was generated randomly so that the average separation

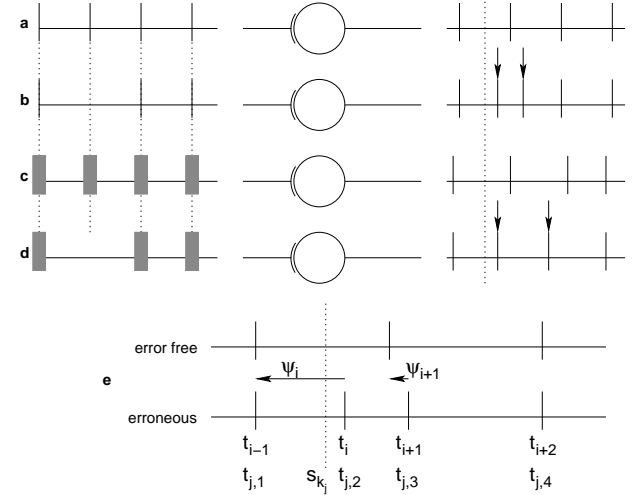


Fig. 2. Input generation and output perturbation. A sequence of input PSPs (left column) is presented via an inhibitory synapse to the SAO (middle column), producing a sequence of spikes, or APs (right column). High precision (a and b) and low precision timing (c and d) were used, both without (a and c) and with (b and d) errors (missing PSPs). Errors result in perturbed outputs (a–d, right column, arrows). Perturbation is measured by computing cross intervals between APs just after an error (e, bottom train) and APs in the matching error-free simulation (e, top train).

between errors was $\gg I$; this allowed the system to return to its nominal (error free) state between errors. This was then used to produce a high temporal precision train with errors, $\mathcal{H}_e = \mathcal{H} - \mathcal{S}$ (b, left), termed an *erroneous train*.

A number of low precision PSP trains were then generated. These trains were intended to be possible low precision counterparts of the canonical train, illustrated in Figure 2(c) by the grey bars. In such a train, the presence of a PSP at any time during some window (width determined by the amount of temporal precision) would be considered to convey the same information. “Jitter” values u_k , taken from a uniform distribution with range $\pm wI$, $w < 1$, were used to alter the time of each canonical PSP to produce a low precision train, $\mathcal{L} = \{s_1 + u_1, s_2 + u_2, \dots, s_n + u_n\}$. The value of w was selected so that it would not alter the model’s dynamical behavior: in this case, that meant that the result was an “imprecise”, or “noisy” 1:1 locking. The same errors as used in the high precision case were then introduced — the same PSPs \mathcal{K} were removed — to produce low precision erroneous trains \mathcal{L}_e (d, left).

Analysis focused on the times just before and after errors in matched pairs of simulations ($\mathcal{H}, \mathcal{H}_e$) or ($\mathcal{L}, \mathcal{L}_e$). Because of the comparatively long separation between errors, it was expected that the model’s postsynaptic discharge (APs, right column in Figure 2) would be identical just before the error for both of the matching error-free and erroneous train pairs. This was verified by the analysis (see Results). For each error time $s_{k_j} \in \mathcal{S}$, the few APs just before and after were compared to those in the error-free train to determine the perturbation induced by the missing PSP. This comparison, shown in Fig-

ure 2(e), was done by plotting the temporal difference (cross interval) ψ_i between the AP in the response to \mathcal{H}_e or \mathcal{L}_e at time t_i and the AP in the response to \mathcal{H} or \mathcal{L} at time $\max_{t'_i \leq t_i} t'_i$; the “matching” AP in the error-free case. This was then plotted versus the time from error occurrence (cross interval $t_i - s_{k_j}$) to produce a *perturbation plot*. The erroneous train’s APs were used as the reference because the inhibitory nature of the PSPs meant that errors caused *postsynaptic acceleration*. As a result, the erroneous train had more APs than the error-free train; using these APs as reference events assured that none would be missed in the analysis.

A perturbation plot can be used directly to judge recovery time in the high precision case, as each error produced identical responses. The perturbation points $p_i^{\mathcal{H}, \mathcal{L}} = (t_i - s_{k_j}, \psi_i)$ were used to compare perturbation recovery time for high ($p^{\mathcal{H}}$) versus low ($p^{\mathcal{L}}$) precision inputs. The same number of points $p_i^{\mathcal{L}}$ were in all low precision perturbation plots; this was also the number of $p_i^{\mathcal{H}}$ in the high precision one. Each of these points was assigned to one of c clusters, based on $t_i - s_{k_j}$, corresponding to the AP just before an error, the AP preceding that one, etc., and the AP just after an error, the AP after that one, etc. From error to error, these AP times will vary somewhat, but that variation is much less than the interspike interval, and so $t_i - s_{k_j}$ will be tightly clustered around c values (c being selected when making the plot). Therefore, for each error k_j at time s_{k_j} ($j = 1, 2, \dots, m$), there will be c APs, occurring at times $(t_{j,1}, t_{j,2}, \dots, t_{j,c})$. This is illustrated in Figure 2(e) for $c = 4$, where $t_{j,1} = t_{i-1}$, $t_{j,2} = t_i$, $t_{j,3} = t_{i+1}$, and $t_{j,4} = t_{i+2}$. Conversely, for a cluster g ($g = 1, 2, \dots, c$), there are m APs, one corresponding to each error (at times S).

Using this nomenclature, the perturbation that corresponds to an AP at time $t_{j,g}$ (the g^{th} AP around error number j) is $\psi_{j,g}^{\mathcal{H}}$ (for high precision inputs) or $\psi_{j,g}^{\mathcal{L}}$ (for low precision inputs). Let us assign to cluster g a single time $e_g = \overline{t_{j,g} - s_{k_j}}$ as the mean of the offset of each spike in that cluster (over all of the simulations — high and low precision — taken together) from its corresponding error. We can then compute the difference $\delta_{j,g} = \psi_{j,g}^{\mathcal{L}} - \psi_{j,g}^{\mathcal{H}}$ for each low precision AP, yielding the set $\delta_g = \{\delta_{1,g}, \delta_{2,g}, \dots, \delta_{m,g}\}$ for all APs in a cluster. A *recovery plot* is a plot of the range of δ_g , $\Delta_g = \max(\delta_g) - \min(\delta_g)$, versus cluster time e_g . The low precision train was considered to be fully recovered from an error when the Δ_g returned to its pre-error value.

III. RESULTS

Figures 3 through 6 present basic plots and return maps for high precision (3 and 4) and low precision (5 and 6) PSP trains. The presynaptic interval (exact for high precision, average for low precision) was $I = 0.186456s$, which, for a natural interval $N = 0.104415s$, yielded $N/I = 0.56$. The high precision plots clearly show a 1:1 locking behavior, with the $T_i = I$ upon application of the PSP train. For the low precision train, the inaccuracy was $w = 0.01$, yielding $-0.00186 \leq u_k \leq 0.00186$.

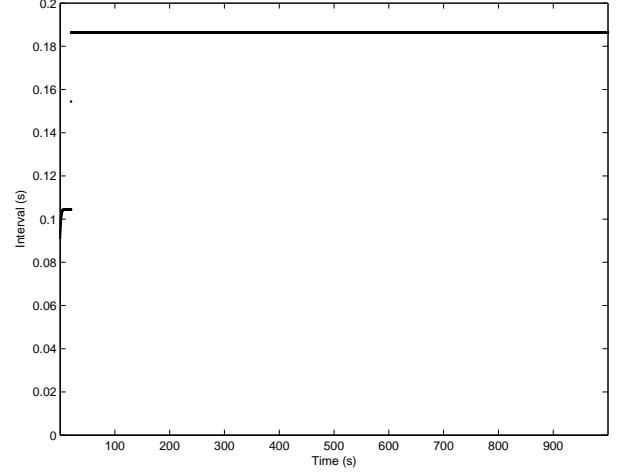


Fig. 3. Postsynaptic interval basic plot, high precision PSP train starting at 20s.

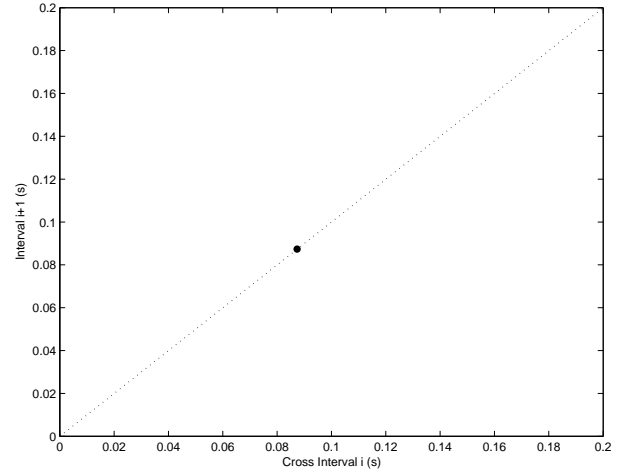


Fig. 4. Phase return map, high precision PSP train, for all APs after 30s.

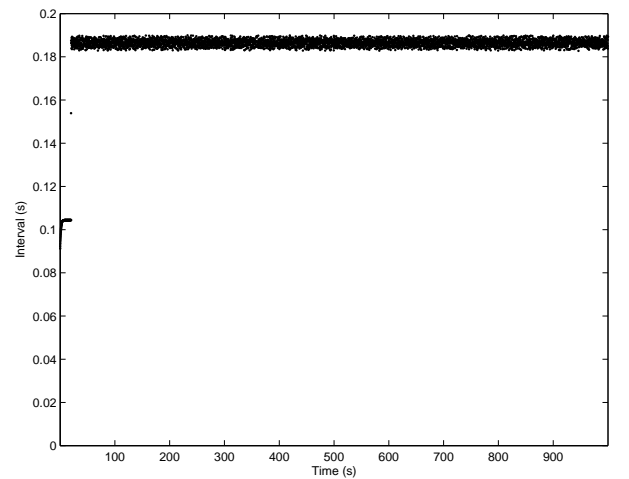


Fig. 5. Postsynaptic interval basic plot, low precision ($\pm 1\%$) PSP train starting at 20s.

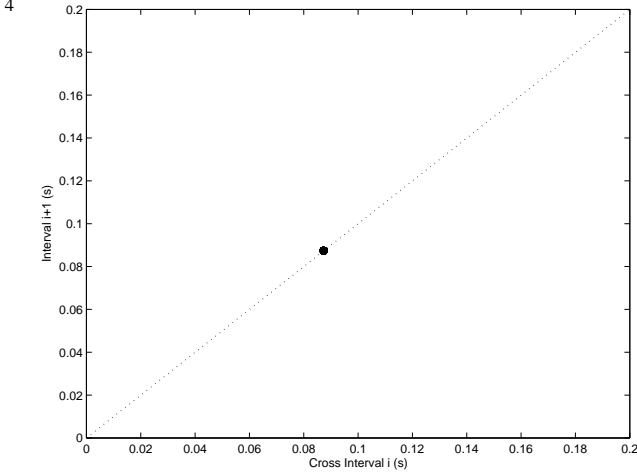


Fig. 6. Phase return map, low precision ($\pm 1\%$) PSP train, for all APs after 30s.

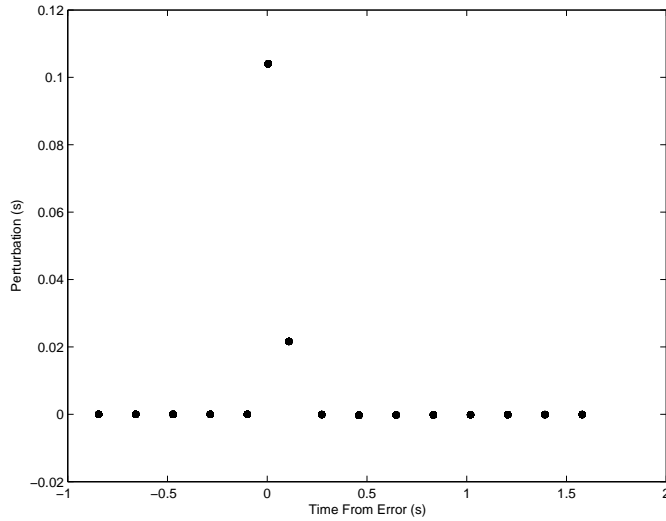


Fig. 7. Perturbation plot for high precision input. Number of missing PSPs, $m = 98$.

Figure 5 shows that, over the entire 1000s simulation, postsynaptic intervals resembled presynaptic ones. That this postsynaptic variation was caused by the presynaptic jitter is verified by Figure 6, which shows that cross intervals (phases) between APs and PSPs were almost invariant. This means that variation in the response of the SAO directly followed variation in its input: each PSP substantially reset the neural oscillator's phase. All further results are based on this high precision simulation and ten low precision simulations with identical parameters.

Errors were generated so that at least 5 seconds separated each. Figure 7 is a perturbation plot for a high precision situation, $(\mathcal{H}, \mathcal{H}_e)$. As described in Methods, this graph plots cross intervals between APs in the erroneous and error-free trains around the time of the missing PSP versus time relative to that of the missing PSP. So, zero on the X-axis is the time of the missing PSP s_{k_j} and all cross intervals around all errors are

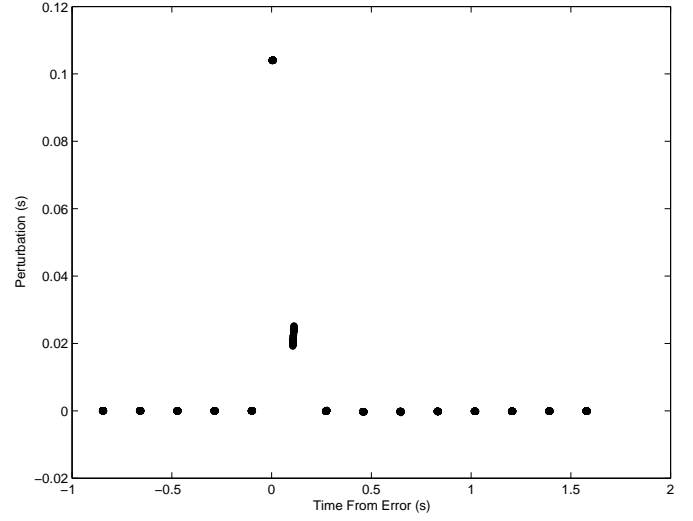


Fig. 8. Perturbation plot for low precision input. Number of missing PSPs, $m = 98$.

superimposed, or *pooled*, to produce the graph.

Despite the pooling of multiple perturbations, there is no noticeable spread among the points plotted, supporting the conclusion that each missing PSP produced an identical perturbation in the high-precision situation. Additionally, all points have a perturbation of 0 just before $\tau = s_{k_j}$ indicates that the model was firing at identical times in both the error-free and erroneous cases. Absence of a single inhibitory PSP allowed the model to fire earlier than it otherwise would have — only a bit over 0.1s after the preceding AP — and in fact faster than its natural interval. This is the well-known post-inhibitory rebound. The second APs are also perturbed; in fact, the erroneous trains examined here always produced one more PSP than the error-free ones after each error.

For comparison purposes, Figure 8 pools the perturbation recovery for low precision input for 98 errors. Here, there is little dispersal of the time of the first AP after an error (near $t = 0$) and noticeable dispersal of the time of the second AP after an error (on the order of 10ms).

Next, the aftereffects of the error were compared between the multiple low precision trains and the high precision train. The recovery plot is presented in Figure 9. This displays the range of differences in perturbations between the low and high precision cases for all errors in all ten low precision trains (a total of 980 errors over 10,000s simulation time). Up until the error time, the difference was negligible (around $5\mu\text{s}$); it is assumed that this represents a sufficient test of significant perturbation difference. Note that the range of perturbation differences for the first post-error APs was 0.16ms, while the second's was much larger at 7.2ms. Thereafter, there was a short period of "ringing", and then an approximately exponential decay with a time constant of 1.15s. As a result, there could be a significant perturbation difference for twenty or more APs — more than 3.5s.

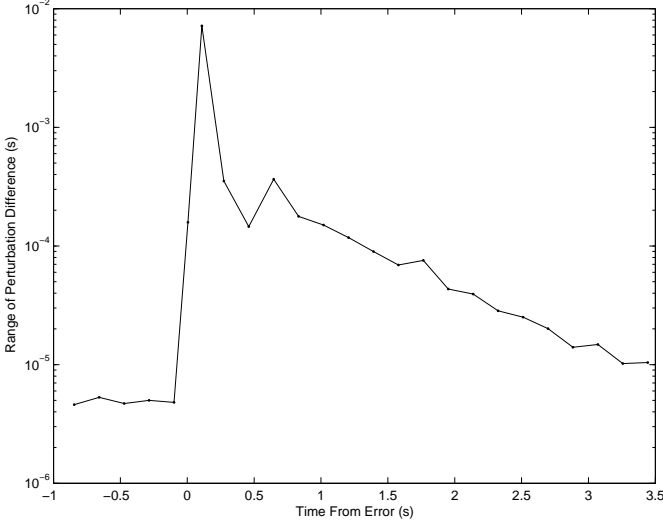


Fig. 9. Comparison of perturbation recovery in low and high precision simulations.

IV. DISCUSSION

Preliminary results have been presented that support the hypothesis that high temporal precision in a presynaptic spike train could be used as an error correction mechanism. Errors were missing PSPs, and error correction was determined by how quickly the neuron returned to its error-free firing times.

Figure 9 plots the *range* of differences between all low and high precision error responses. Those responses are the difference in AP timing between simulations identical *in every way*, except for the one missing PSP at time s_{k_j} . This is evidenced by all trains having APs at almost precisely the same times just before the missing PSP.

These simulations used powerful PSPs, each greatly resetting the neuron's state. This is apparent in Figures 8 and 9: the first AP after an error had much smaller variability than the low precision PSP times (0.16ms versus 3.8ms). So, regardless of the varying history of preceding PSPs, the time of the next AP was mostly determined by the most recent PSP's timing. This suggests that there is a fast time constant of post-inhibitory rebound that is mostly governed by the cell's own internal dynamics.

The second AP after an error (which occurred *after* the first "post-error" PSP) had variability on the order of twice the PSP inaccuracy (7.2ms versus 3.8ms). This indicates that PSPs don't completely reset the cell's state — that the memory of previous PSP variation shows through at a somewhat longer time constant. Additionally, that fact that the low precision trains had a range of perturbed responses thereafter was *not* due to differences in PSP times after the error, as PSPs occurred at exactly the same times for erroneous and error-free simulations. This range of perturbations was, instead, also a long-term aftereffect of different histories of PSP times before the *different* errors. In other words, the time it took to recover from an error depended on the timing of more than one PSP before the error.

The durability of internal neuronal state is key to both error correction and computation. In the presence of stationary input, a neuron's state will evolve along some *attractor* in its state space. High precision input will result in a state that closely follows this attractor; the variation in low precision input moves the state off the attractor. In effect, the neuron's state in the low precision case is a "ball", rather than a point, with the size of the ball determined by the input imprecision. Because the state is likely to be away from the attractor, it quite often takes longer to return to the attractor's vicinity after a perturbation (error). How much longer depends on the gradient of the attractor basin.

The simulations performed here are viewed as an especially demanding test of the hypothesis. The dynamical behavior selected, phase locking, is the most robust in the presence of presynaptic variability [10]. The moderately powerful PSPs used almost completely reset the neuronal oscillator's state. Additionally, only a small degree of imprecision ($\pm 1\%$) was used. Therefore, it is expected that this can serve as a reasonable lower bound on the amount of error correction possible with highly precise temporal coding in spike trains.

APPENDIX

The model was initially developed for the lobster SAO and FAO (fast adapting stretch receptor organ) [9] and modified to match the pacemaking behavior of the crayfish SAO (using a bias current (1) and to include an inhibitory synapse (6). Two ionic fluxes, Na^+ and K^+ , are controlled by voltage-sensitive permeabilities, P_{Na} and P_{K} (2, 3). Three leakage permeabilities, P_{LNa} , P_{LK} , and P_{LCl} , present constant pathways for Na^+ , K^+ , and Cl^- fluxes, respectively (4). Two active pumping pathways, I_{pNa} and I_{pK} work to maintain internal ionic concentrations (5). The membrane also has an associated capacitance, C_m (1).

Presynaptic inputs cause fixed-duration changes in the synaptic permeability, P_{syn} , to Cl^- [20]. The membrane potential equation is presented in equation (1), where the currents I_X (for type X) are computed from ionic fluxes as shown in (2–6).

$$\frac{dV_m}{dt} = -(I_{\text{Na}} + I_{\text{K}} + I_{\text{LNa}} + I_{\text{LK}} + I_{\text{LCl}} + I_{\text{p}} + I_{\text{bias}} + I_{\text{syn}})/C_m \quad (1)$$

$$I_{\text{Na}} = A\bar{P}_{\text{Na}}m^2hl \frac{V_m F^2}{RT} \times \frac{[\text{Na}^+]_o - [\text{Na}^+]_i \exp(FV_m/RT)}{1 - \exp(FV_m/RT)} \quad (2)$$

$$I_{\text{K}} = A\bar{P}_{\text{K}}n^2r \frac{V_m F^2}{RT} \times \frac{[\text{K}^+]_o - [\text{K}^+]_i \exp(FV_m/RT)}{1 - \exp(FV_m/RT)} \quad (3)$$

$$I_{\text{LX}} = AP_{\text{LX}} \frac{V_m F^2}{RT} \times \frac{[X]_o - [X]_i \exp(FV_m/RT)}{1 - \exp(FV_m/RT)} \quad (4)$$

$$I_p = \frac{AF}{3} \frac{\bar{J}_{p,Na}}{\left(1 + \frac{K_m}{[\text{Na}^+]_i}\right)^3} \quad (5)$$

$$I_{syn} = A\bar{P}_{syn} \frac{V_m F^2}{RT} \times \frac{[\text{Cl}^-]_o - [\text{Cl}^-]_i \exp FV_m/RT}{1 - \exp FV_m/RT} \times \sum_{\forall s_k < t} (e^{(s_k-t)/\tau_+} - e^{(s_k-t)/\tau_-}) \quad (6)$$

Here A is the cell membrane area, \bar{P}_X is the maximum permeability to ion X , m , h , l , n , and r are gating variables, T is absolute temperature, and F and R are the Faraday and universal gas constants. The pumping mechanism exchanges 3 Na^+ ions for 2 K^+ in (5), where $\bar{J}_{p,Na}$ is the maximum Na^+ pump capacity and K_m is a constant. $[\text{Cl}^-]_i$, $[\text{K}^+]_i$, and $[\text{Na}^+]_i$ are assumed constant. The gating variables' kinetics are defined by (7) (where $g \in \{m, h, l, n, r\}$), with asymptotic values g_∞ and time constants τ_g given by (8) and (9).

$$dg/dt = (g_\infty - g) \frac{1}{\tau_g} \quad (7)$$

$$g_\infty = \nu_g + \frac{1 - \nu_g}{1 + \exp \left[\frac{z_g e}{kT} (V_m - V_g) \right]} \quad (8)$$

$$\tau_g = Q_g \bar{\tau}_g / \left\{ \exp \left[\frac{\delta_g z_g e}{kT} (V_m - V_g) \right] + \exp \left[\frac{(\delta_g - 1) z_g e}{kT} (V_m - V_g) \right] \right\} \quad (9)$$

$$Q_g = \left(\frac{1 - \delta_g}{\delta_g} \right)^{\delta_g} + \left(\frac{1 - \delta_g}{\delta_g} \right)^{\delta_g - 1} \quad (10)$$

ν_g is the minimum relative channel permeability, z_g the effective valency of the gating structure, e the electron charge, V_g the membrane voltage at which half of the gating system is in its "open" state, $\bar{\tau}_g$ the maximum value of τ_g , and δ_g the degree of energy barrier asymmetry (between 0 and 1).

ACKNOWLEDGMENT

The author would like to thank Patricia Tressel for her suggestions, which greatly improved this manuscript.

REFERENCES

- [1] W. Bialek, F. Rieke, R. R. de Ruyter van Steveninck, and D. Warland, "Reading a neural code," *Science*, vol. 252, pp. 1854–7, June 1991.
- [2] A. Borst and F. Theunissen, "Information theory and neural coding," *Nature Neurosci.*, vol. 2, pp. 947–57, Nov. 1999.
- [3] G. Fuhrmann, I. Segev, H. Markram, and M. Tsodyks, "Coding of temporal information by activity-dependent synapses," *J. Neurophysiol.*, vol. 87, pp. 140–8, Jan. 2002.
- [4] Y. Dan, J.-M. Alonso, W. M. Usrey, and R. C. Reid, "Coding of visual information by precisely correlated spikes in the lateral geniculate nucleus," *Nature Neurosci.*, vol. 1, pp. 501–7, Oct. 1998.

- [5] R. Lestienne, "Spike timing, synchronization and information processing on the sensory side of the central nervous system," *Prog. Neurobiol.*, vol. 65, pp. 545–591, 2001.
- [6] R. Wessel, C. Koch, and F. Gabbiani, "Coding of time-varying electric field amplitude modulations in a wave-type electric fish," *J. Neurophysiol.*, vol. 75, pp. 2280–93, June 1996.
- [7] J. Segundo, J.-F. Vibert, K. Pakdaman, M. Stiber, and O. Diez Martínez, "Noise and the neurosciences: a long history, a recent revival and some theory," in *Origins: Brain and Self Organization* (K. Pribram, ed.), Lawrence Erlbaum Assoc., in press, 1994.
- [8] F. Theunissen, J. C. Roddey, S. Stufflebeam, H. Clague, and J. P. Miller, "Information theoretic analysis of dynamical encoding by four identified primary sensory interneurons in the cricket cercal system," *J. Neurophysiol.*, vol. 75, pp. 1345–64, Apr. 1996.
- [9] A. Edman, S. Gestrelus, and W. Grampp, "Analysis of gated membrane currents and mechanisms of firing control in the rapidly adapting lobster stretch receptor neurone," *J. Physiol.*, vol. 384, pp. 649–69, 1987.
- [10] M. Stiber, R. Jeong, and J. Segundo, "Responses to transients in living and simulated neurons," *IEEE Trans. Neural Networks*, vol. 8, no. 6, pp. 1379–85, 1997.
- [11] M. Stiber, K. Pakdaman, J.-F. Vibert, E. Boussard, J. Segundo, T. Nomura, S. Sato, and S. Doi, "Complex responses of living pacemaker neurons to pacemaker inhibition: a comparison of dynamical models," *Biosystems*, vol. 40, pp. 177–88, 1996.
- [12] A. Hindmarsh, "ODEPACK: A systematized collection of ODE solvers," in *Scientific Computing* (R. Stepleman, ed.), pp. 55–64, Amsterdam: North-Holland, 1983.
- [13] I. The MathWorks. <http://www.matworks.com/>.
- [14] J. P. Segundo, E. Altshuler, M. Stiber, and A. Garfinkel, "Periodic inhibition of living pacemaker neurons: I. Locked, intermittent, messy, and hopping behaviors," *Int. J. Bifurcation and Chaos*, vol. 1, pp. 549–81, Sept. 1991.
- [15] J. P. Segundo, E. Altshuler, M. Stiber, and A. Garfinkel, "Periodic inhibition of living pacemaker neurons: II. Influences of driver rates and transients and of non-driven post-synaptic rates," *Int. J. Bifurcation and Chaos*, vol. 1, pp. 873–90, Dec. 1991.
- [16] J. Segundo, M. Stiber, and J.-F. Vibert, "Synaptic coding of spike trains. Entrainment across synapses of one neuron by another," in *The Handbook of Brain Theory and Neural Networks* (M. Arbib, ed.), pp. 953–6, MIT Press, 1995.
- [17] G. Sugihara, M. Stiber, L. Bercier, R. Penner, and J. Segundo, "The irregular discharges of inhibited pacemaker neurons: noisy, deterministic, linear and nonlinear issues," in *Int. Workshop on Neuronal Coding*, (Prague), Sept. 1995.
- [18] G. Sugihara, M. Stiber, L. Bercier, R. Penner, and J. Segundo, "The irregular, complex and unpredictable discharges of inhibited pacemaker neurons. dimensions, linearity and non-linearity. noise and deterministic issues," *Neurosci.*, submitted, 1996.
- [19] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press/Bradford Books, 1997.
- [20] K. Ozawa and K. Tsuda, "Membrane permeability change during inhibitory transmitter action in crayfish receptor cell," *J. Neurophysiol.*, vol. 36, no. 5, pp. 805–16, 1973.



Michael Stiber (S'81–M'83) is an Associate Professor in the Computing and Software Systems Program at the Univ. of Washington, Bothell. Previously, he had been a Research Assistant Professor in the Dept. of Molecular and Cell Biology at the Univ. of Calif., Berkeley and an Assistant Professor in the Dept. of Computer Science at the Hong Kong University of Science and Technology. Dr. Stiber received a B.S. in computer science and a B.S. in electrical engineering from Washington Univ., St. Louis, MO in 1983 and the M.S. and Ph.D. degrees in computer science from UCLA in 1990 and 1992, respectively. He has held positions with Texas Instruments and Philips, and is a frequent visitor to the Department of Biophysical Engineering at Osaka University, Japan. His research interests include neural networks, computational biology, bioinformatics, scientific data management and knowledge base systems, and scientific visualization. Dr. Stiber is a member of the Association for Computing Machinery, Tau Beta Pi, and Eta Kappa Nu. He is also an officer in the IEEE Computer Society Seattle section.