Computing with recurrent spiking networks: from mathematical optimality towards biological plausibility

Jorge Aurelio Menéndez

CoMPLEX Mini-Project #2

Supervisor: Peter E. Latham

14th March 2016

Contents

1. Introduction

One of the most difficult and important challenges facing neuroscience is the question of how neurons compute. The brain consists largely of interconnected neurons that communicate via discrete action potentials - an interaction that somehow gives rise to the variety of computations that the brain performs on a millisecond-by-millisecond basis. [10] How this is achieved is not yet understood, but many efforts have been made recently to construct model neural networks of spiking neurons that can perform basic computations [2, 21, 15, 3] (for a review, see [5]).

Such models - which I will call *spiking network* models - differ markedly from previous neural network models in several respects. Firstly, there is the obvious difference that neurons in spiking network models output discrete spikes rather than continuous values. This contrasts with the classical connectionist approach [14, 18] as well as with more modern rate-coding approaches. [6] The result, of course, is more biologically plausible models of neural circuitry, since it is known that neurons don't communicate via firing rates. Rather, a neuron passes information onto another via synaptic transmission of discrete action potentials.

That said, neural firing rates can still contain important information. While neurons in a network may only communicate with each other by way of discrete spikes, the resulting individual firing rates may encode the information represented by the network. [6, 23] Indeed, the great variability in spiking observed in cortical recordings suggests that this may be the case: a neuron that spikes 10 times in one second can be encoding the same thing as a neuron that spikes at 10 different times within that same second. [16, 11] A rate-based code would thus explain the commonly observed trial-to-trial variability in spike times in response to the same stimulus. [22]

Rate-coding does not come without its problems, however. Firstly, any temporal information about the individual presynaptic events preceding an action potential is discarded by a rate-code, which averages them out. Even if we assume such information is irrelevant to the computation at hand, the precision of a rate code under the stochastic regime in which neural firing seems to operate is constrained by the number of spikes used. In other words, for high precision you need either a large populuation of neurons or high firing rates. [5] On the mechanical side of things, it turns out that random Poisson-like pre-synaptic spike trains generate unrealistic regular spike trains in the post-synaptic neuron [19], which is particularly problematic given the insight that most of neural response variability originates in its synaptic inputs. [12]

The rate-coding solution to this latter problem is to balance the excitatory and inhibitory inputs to a neuron. [16, 23] But we the first two problems remain, and we end up with a less rich and more inefficient code. Spiking networks can provide a solution to all three problems, by balancing excitation and inhibition and ensuring the neurons only spike when they have to. [5]

Below, we examine the properties and mechanisms of such networks, with the aim of assessing and possibly improving their biological plausibility. I begin by deriving the architecture and dynamics of a prototypical predictive coding spiking network, in the vein of Boerlin *et al* (2013). I then go on to simulate a slightly more biologically realistic variation on the model and test the effect of several modifications to it. Finally, I derive the structure of a rate-coding network with similar functionality and discuss its relevance.

2. Derivation of the Spiking Network Model

We take the approach of Boerlin *et al* (2013) to constructing a network of spiking neurons that represents a J-dimensional variable $\mathbf{x}$ with dynamics

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{c}(t) \tag{1}$$

The goal is to derive the dynamics of the individual neuron membrane potentials such that at any time t, the network's representation $\hat{\mathbf{x}}(t)$ is an accurate estimate of $\mathbf{x}(t)$. As emphasized above, we take a spike-based coding approach wherein the

network's representation at time t is a function of all the individual neuron spike trains at that time. Thus, we aim to derive dynamics that will produce spike trains yielding a representation $\hat{\mathbf{x}}$ with dynamics approximately equal to $\frac{d\mathbf{x}}{dt}$.

The derivation springs from three key assumptions. Firstly, we model the network as a set of neurons with recurrent connections that take an external feedforward input $\mathbf{c}(t)$. Thus, we view equation 1 as the dynamics of a variable with a state transition matrix $\mathbf{A}$ and command variables $\mathbf{c}(t)$. Whereas the transition matrix is built into the recurrent connections, the commands are injected into the network as external inputs. This terminology highlights the relevance of the computation carried out by this network to control theory. [2]

Secondly, we assume that the representation of the network, which I will refer to as the *network estimate* $\hat{\mathbf{x}}$, has the following dynamics:

$$\frac{d\hat{\mathbf{x}}}{dt} = -\lambda_d \hat{\mathbf{x}} + \mathbf{\Gamma o} * \kappa(t) \tag{2}$$

where κ is an arbitrary synaptic kernel (e.g. a δ function or a decaying exponential), λ_d is a leak rate, $\mathbf{o}$ is a vector of spike trains $o_i(t) = \sum_k \delta(t - t_i^k)$ for each of the $i = 1, ..., N$ neurons, and $\mathbf{\Gamma}$ is a $J \times N$ matrix providing the weights of the contributions of each of the N neurons to the network estimate of each of the J components of $\mathbf{x}$. We call each of the $i = 1, ..., N$ columns of $\mathbf{\Gamma}$, henceforth notated as $\mathbf{\Gamma}_i$, the *decoding kernel* of neuron i. Solving equation 2, we get that the network estimate is a weighted and leaky sum over spike trains $\hat{\mathbf{x}} = \mathbf{\Gamma o} * \kappa * h_d(t)$, where $h_d(t) = \Theta(t)e^{-\lambda_d t}$ is a decaying exponential kernel.

Lastly, we assume that an arbitrary neuron i will spike at time t if and only if this yields a decrease in the squared network estimate error $E(t)$:

$$E(t) = \int_0^t d\tau (\mathbf{x}(\tau) - \hat{\mathbf{x}}(\tau))^2 \tag{3}$$

Noting that a spike from i changes the estimate $\hat{\mathbf{x}}$ by adding a synaptic kernel weighted by the decoding kernel of neuron i ($\hat{\mathbf{x}}(t) \to \hat{\mathbf{x}}(t) + \mathbf{\Gamma}_i \mathbf{o} * \kappa(\epsilon)$), our spiking condition for neuron i becomes:

$$\int_0^{t+\epsilon} d\tau (\mathbf{x}(\tau) - \hat{\mathbf{x}}(\tau) - \mathbf{\Gamma}_i \mathbf{o} * \kappa(\tau))^2 < \int_0^{t+\epsilon} d\tau (\mathbf{x}(\tau) - \hat{\mathbf{x}}(\tau))^2 \tag{4}$$

where ϵ is some amount of time into the future over which we minimize the error. For simple cases (e.g. when $\kappa(t) = \delta(t)$), a "greedy" minimization of $E(t)$ may be performed where $\epsilon = 0$. [2, 3] For others, however (e.g. the difference-of-exponentials synapse case [15], see section 3.7 below) a small $\epsilon > 0$ may be required. Carrying out the integration and doing some algebra, equation 4 gives us

$$\mathbf{\Gamma}_i^T (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) > \frac{\|\mathbf{\Gamma}_i\|^2 \kappa(\epsilon)}{2} \tag{5}$$

This is the spiking condition for neuron i: when equation 5 is satisfied, neuron i should spike.

The task is now to define individual neural dynamics such that this happens. This turns out to be extremely simple: equate the left side of equation 5 to the membrane potential of the ith neuron V_i and the right side to i's spiking threshold T_i. We now have an equation for the membrane potentials of each of the neurons in the network that will lead to spikes in accordance to our spiking rule:

$$\mathbf{V}(t) = \mathbf{\Gamma}^T(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \tag{6}$$

Put intuitively, the membrane potential of the ith neuron is the network estimate's error projected onto i's decoding kernel.

From here, we can derive the individual neuron dynamics:

$$\frac{d\mathbf{V}}{dt} = \mathbf{\Gamma}^T(\frac{d\mathbf{x}}{dt} - \frac{d\hat{\mathbf{x}}}{dt}) \tag{7}$$

Replacing the dynamics for $\mathbf{x}$ and $\hat{\mathbf{x}}$ with equations 1 and 2, respectively:

$$\frac{d\mathbf{V}}{dt} = \mathbf{\Gamma}^T((\mathbf{A}\mathbf{x} + \mathbf{c}) - (-\lambda_d\hat{\mathbf{x}} + \mathbf{\Gamma}\mathbf{o} * \kappa)) \tag{8}$$

Assuming the network estimate is a good approximation of $\mathbf{x}$, we can then replace $\mathbf{x}$ with $\hat{\mathbf{x}}$:

$$\frac{d\mathbf{V}}{dt} = \mathbf{\Gamma}^T\mathbf{c} + \mathbf{\Gamma}^T(\mathbf{A} + \lambda_d\mathbf{I})\hat{\mathbf{x}} - \mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{o} * \kappa \tag{9}$$

where $\mathbf{I}$ is the identity matrix. Expanding $\hat{\mathbf{x}}$ to its components and adding a generic leak term for biological plausibility [10]:

$$\frac{d\mathbf{V}}{dt} = -\lambda_V\mathbf{V} + \mathbf{\Gamma}^T\mathbf{c} + \mathbf{\Gamma}^T(\mathbf{A} + \lambda_d\mathbf{I})\mathbf{\Gamma}\mathbf{o} * \kappa * h_d - \mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{o} * \kappa \tag{10}$$

We see find that our final dynamics contain four terms: (1) a leak term, (2) external input, (3) "slow" excitatory recurrent connections, and (4) "fast" inhibitory connections. We loosely call these latter two terms "slow" and "fast", respectively, because the "slow" ones are slowed down by convolution with the decaying exponential h_d. We assume that the synaptic kernel κ is either a δ-function (as in [2]) or some flavour of quickly rising and decaying function (such as a decaying exponential [15] or a difference-of-exponentials /citechalk2015) that reflects the transient nature of post-synaptic potentials.

In the simplest case, $\kappa(t) = \delta(t)$, and you get exactly the network derived in Boerlin *et al* (2013). Here, the "fast" inhibitory autapses are particularly important because they implement the membrane potential reset after an action potential (which is not at all biologically realistic). In the case where all neurons have equal decoding kernels (i.e. $\forall i, j\, \mathbf{\Gamma}_i = \mathbf{\Gamma}_j$), it can be shown that, in the limit of high firing rates, the network estimate will have dynamics $\dot{\hat{x}} = \dot{x} + \Gamma_i\lambda_V/2$, thus providing a good estimate of a one-dimensional signal $x(t)$ as long as λ_V is small (remember that the parameter λ_V was introduced for biological plausibility, it did not follow from the derivation). [2]

We note, however, that this model violates Dale's law, as each neuron needs to send excitation and inhibition to its post-synaptic neurons (including itself). It turns out this problem can be circumvented by producing two separate populations of inhibitory and excitatory neurons, and deriving the inhibitory neuron dynamics such that they closely track the excitatory population estimate. In doing so, the inhibitory neurons effectively implement the inhibitory signals necessary for equation 10 to hold. See Appendix A for the derivation. The full network architecture is schematized in figure 1.

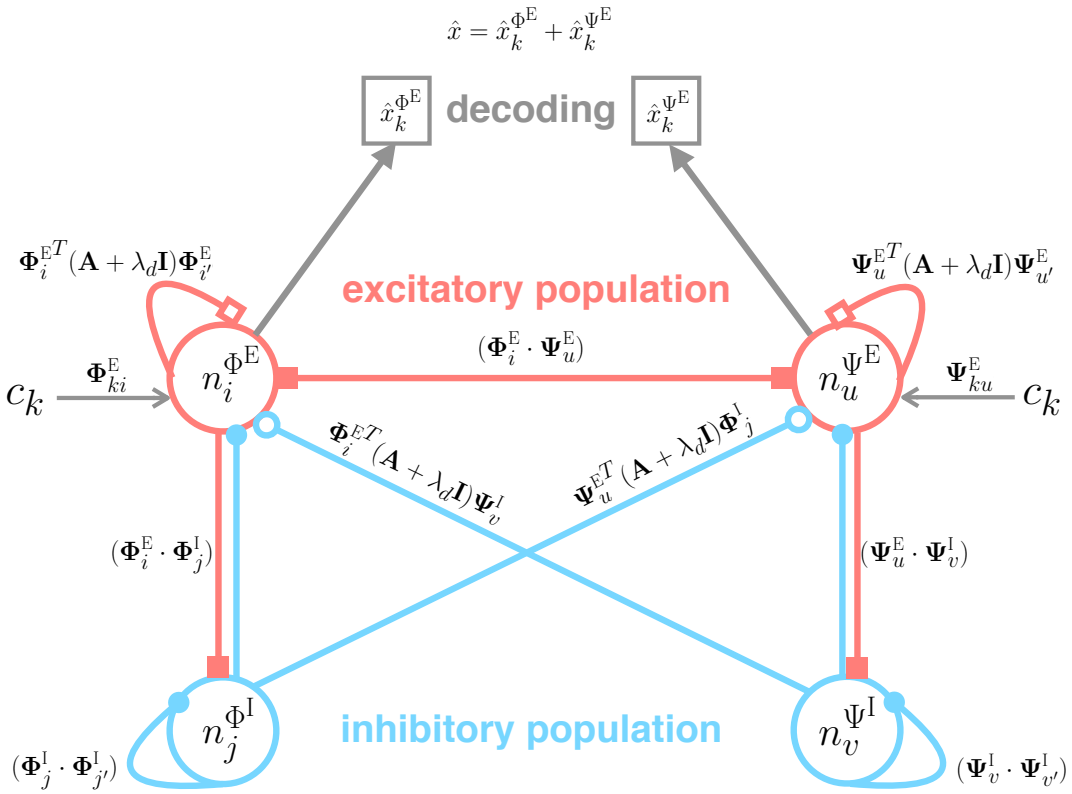

FIGURE 1. Schematic of the network architecture we consider. $\mathbf{\Phi}$ and $\mathbf{\Psi}$ denote the matrices of positive and negative decoding kernels, respectively, where $\mathbf{\Phi}_i$ is the decoding kernel of the ith neuron with a positive kernel (i.e. it is the ith column of $\mathbf{\Phi}$) and $\mathbf{\Psi}_u$ is the decoding kernel of the uth neuron with a negative kernel. E and I denote excitatory and inhibitory neurons, indexed by i (for positive kernels) or u (negative) and j (positive) or v (negative), respectively. Blue open and closed circles denote "slow" and "fast" inhibitory synapses and the same goes for the red squares representing excitatory synapses. For example, the ith excitatory neuron with a positive kernel (neuron $n_i^{\Phi^E}$) takes "slow" excitatory synaptic input from all excitatory neurons i' with positive kernels (including itself), "fast" excitatory synaptic input from all excitatory neurons u with negative decoding kernels, "slow" inhibitory synaptic input from inhibitory neurons with negative kernels, and "fast" inhibitory synaptic input from all inhibitory neurons with positive decoding kernels. Note that many of the "fast" connections consist of dot products between decoding kernels, such that neurons with similar decoding kernels elicit stronger excitation/inhibition between each other. $\mathbf{A}$, λ_d, $\hat{x}$, and c, as used in the text.

2.1. **Intuitive explanation of the network dynamics.** What the derived dynamics effectively do is ensure that a neuron spikes iff its contribution to the estimate will reduce the prediction error. Thus the membrane potential of each neuron is equated to the projection of the prediction error onto its decoding kernel. If the prediction error grows in the direction of its kernel, the membrane potential will rise until the neuron eventually fires, in turn adding its kernel to the estimate and cancelling out the error.

But what happens when two neurons have similar kernels? When the error grows in the direction of their kernels, their potentials will simultaneously rise towards threshold. Should both spike, however, the network will overcompensate for its estimate's error. This is where the "fast" inhibitory connections are crucial. When one of these two neurons spikes, it will instantly send inhibition to all neurons with similar kernels. In this manner, the network distributes spiking efficiently across the network: neurons only spike when they are needed. This offers spiking networks a particular advantage over rate networks and makes them extremely robust to synaptic failures and/or lesions. [2, 5]

This mechanism, however, clearly relies on inhibitory signals immediately responding to spikes. When we modify the network architecture to obey Dale's law (Appendix A), we will see that this causes problems with how synaptic transmission is implemented. Furthermore it requires an exquisitely correlated inhibitory and excitatory signal, which is not necessarily empirically supported. [5]

3. Results

We begin by attempting to simulate the Dale's law-obeying spiking network of Boerlin *et al* (2013) derived above, with instant synapses (i.e. with synaptic kernel $\kappa(t) = \delta(t)$). The dynamics of this network cannot be integrated analytically, so we perform numerical integration via Euler's method. Unfortunately, formulating the algorithm for numerical integration in this case turns out to be quite non-trivial. We begin by describing the problem and rationalize our solution, which ends up not exactly implementing the dynamics derived above. We then explore several modifications to our model and show that certain properties of its behaviour persist. Note that we are considering the fully functional Dale's law architecture, in which neurons are either excitatory or inhibitory (E or I) and can have positive (+) or negative (-) kernels (see figure 1). The four subpopulations are thus designated as E+, E-, I+, and I- in the figures below.

3.1. **The spike updating problem.** The most straightforward way of implementing the above model is by numerically integrating the equations for the dynamics of the individual membrane potentials and checking for suprathreshold potentials at the start of every timestep. When a neuron's membrane potential is above threshold, it fires a spike, evoking an instant excitatory/inhibitory

post-synaptic potential (EPSP/IPSP) in all the neurons with "fast" synaptic connections from the spiking neuron. At every timestep, the membrane potentials of the whole the network are updated according to the identity of the suprathreshold spiking neurons, each time adding a spike to those neurons' respective spike trains. The tracking results of this algorithm are plotted in figure 2A.

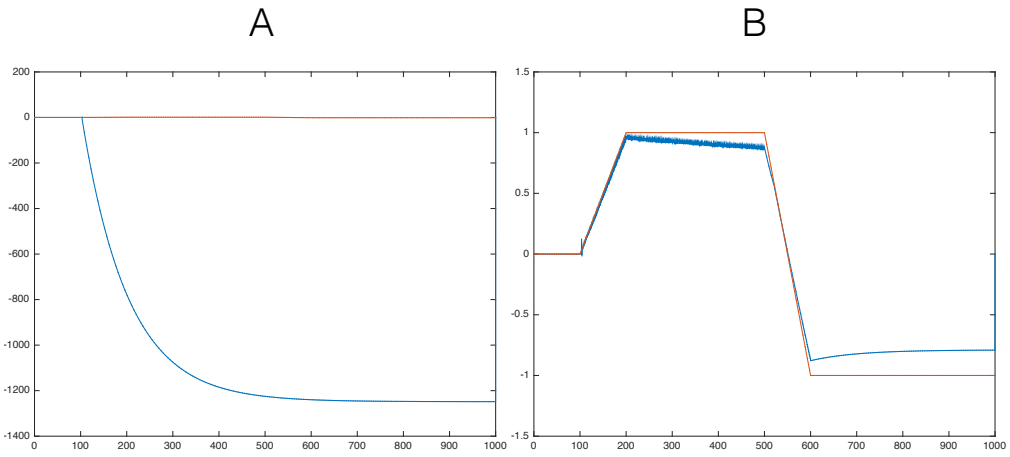

FIGURE 2. Tracking performance of algorithms 1 and 2. Here and in all subsequent similar plots, the orange line denotes the trajectory of the estimate being tracked and the blue line the trajectory of the network estimate. x-axis is in milliseconds, as in all other plots below. Relative error = 1.233×10^3 for A, 0.151 for B.

This implementation of the model clearly leads to great overestimation. In other words, too many excitatory neurons are spiking too often. The culprit of this poor performance is the discrete timesteps inherent to the numerical integration. The issue is appreciated by considering the following scenario: suppose that two excitatory neurons i and i' have membrane potentials above threshold at time t, such that they both spike. Considering the continuous time interval between $t-1$ and t, however, it is likely that one of those neurons would have reached threshold before the other under the exact continuous dynamics. Once one, say i, reached threshold first, the resulting spike would instantly trigger EPSPs in inhibitory neurons that would likely fire, thus instantly triggering IPSPs in the excitatory neurons and bringing the membrane potential of i' back down away from threshold, preventing it from firing anywhere near time t. We can solve this problem by approximating the identity of the spiking neuron in any given timestep by the suprathreshold neuron with the highest membrane potential. Once we impose this constraint on the spike updating algorithm, we get improved performance (figure. 2B).

A problem that remains is that a neuron's membrane potential stays above threshold until an inhibitory neuron fires and evokes an IPSP. This is clearly a

blatant violation of how biological neurons function, where an action potential consists of a spike followed by an immediate fall in membrane potential to below its resting state. Indeed, this issue poses a challenge to the model itself, as it follows directly from the mathematical derivation. We can try to fix this by implementing an explicit voltage self-reset whenever a neuron spikes. While in the mathematical model the reset magnitude is dependent on the identity of the subsequently spiking inhibitory neuron, here we accordingly assume the fixed reset magnitude to be equal to the synaptic weight from the inhibitory neuron most likely to fire following the excitatory neuron's spike, i.e. the one that has the strongest synaptic input from that excitatory neuron. At the time of a spike, we subtract this amount from the current membrane potential of the neuron.[1] The only remaining worry now is that, when an inhibitory spike immediately succeeds an excitatory spike (which almost always occurs), the membrane potential of the spiking excitatory neuron will fall unrealistically low as it will receive a strong IPSP following its self-reset. This is fixed by implementing a kind of reversal potential [4] by imposing a minimum on the membrane potential, set to the spiking threshold minus the self-reset - the minimum potential achievable in the absence of inhibitory input. This algorithm yields the desired performance with a reasonable relative error of .117 (figure 3, top panel).

[1]It might seem more appropriate here to subtract from the spiking threshold, since in continuous time the membrane potential at the time of a spike will be exactly equal to the spiking threshold, by definition. However, the membrane potential would have changed in the time transpired between the time of the spike and the discrete timetstep we are considering, such that, under continuous time, the membrane potential at this timestep would not be equal to the spiking threshold minus the reset. In any case, simulations show that this does not make a difference to network performance. In the subsequently simulated models, we thus subtract the spike reset from the neuron's membrane potential at that time (which will inevitably be above threshold).

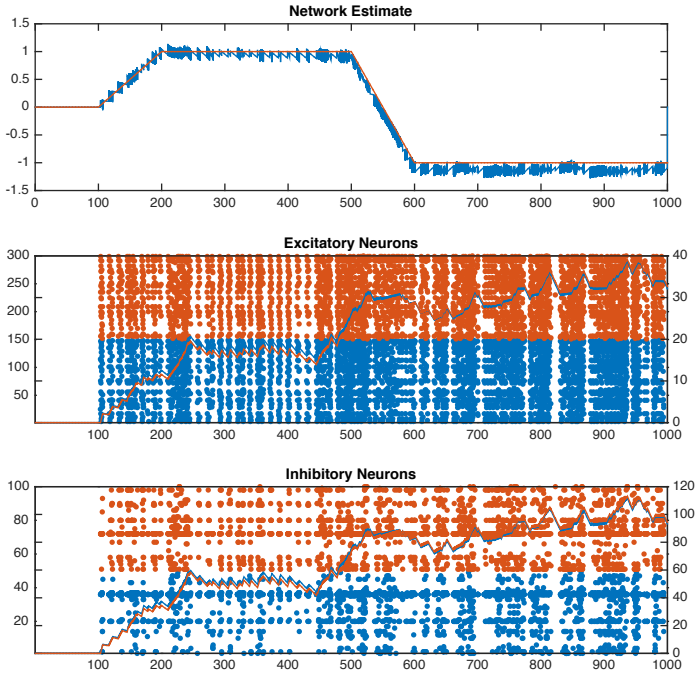

FIGURE 3. Tracking performance and spike rasters from simulating the network with algorithm #3. First panel as above. Second and third panels show spike rasters and population average firing rate timecourses for neurons with positive (blue) and negative (orange) decoding kernels, in the excitatory and inhibitory populations, respectively. For each of these plots, and all similar ones below, the left y-axis denotes neuron index and right y-axis denotes firing rate, in Hz. Tracking error = .117

However, the resulting spike trains from implementing the model with this algorithm are highly correlated and do not resemble the typical irregular patterns observed in real neurons (figure 3, second and third panels). [10, 4] The reason for this might be that, since spike updating occurs simultaneously for all neurons, the large EPSPs in inhibitory neurons following an excitatory spike cannot trigger the subsequent inhibitory spike until the next timestep. As a result, the feedback inhibition that is supposed to silence all other excitatory neurons with similar decoding kernels to the one that just spiked is delayed. Importantly, neurons with similar decoding kernels have correlated membrane potentials, so if one neuron is at or above threshold, then any other neuron with a similar decoding kernel is likely to be as well. Hence, when an excitatory neuron spikes, another neuron with a similar decoding kernel is likely to also be above threshold and thus spike on the next timestep, simultaneously with the feedback inhibition that was supposed to silence it.

We can fix this by simply checking for suprathreshold membrane potentials in the inhibitory population before the excitatory population. Because the IPSPs resulting from the instant synapses between inhibitory and excitatory neurons are quite strong, it is virtually impossible for an excitatory neuron to remain above threshold following an inhibitory spike. Thus, when an inhibitory spike achieves the inhibition it is supposed by lowering the potential of the according excitatory neurons below threshold before the algorithm "spikes" them (i.e. checks for suprathreshold potentials). Furthermore, since the excitatory population is checked after the inhibitory population, when an excitatory spike leads to suprathreshold potentials in a the inhibitory population, the resulting inhibitory spikes are delayed until the next timestep, thus preventing disynaptic transmission within the same timestep (i.e. $i \to j \to i$). Thus, we deviate from the mathematical model by imposing certain constraints to make our model (which is now defined jointly by the dynamics derived above and the spike updating algorithm) more biologically plausible: fixed and automatic self-resets, reversal potentials, and no instant disynaptic transmission.

This algorithm yields the best performance (figure 4) and produces biologically realistic Poisson-like spike trains with population average firing rates in 1-3Hz range. These results resemble those obtained by Boerlin *et al* (2013), who presumably implemented the exact mathematical model via Euler numerical integration.[2] We now go on to examine the properties of our model and explore several variations.

[2]How they did so remains a mystery...

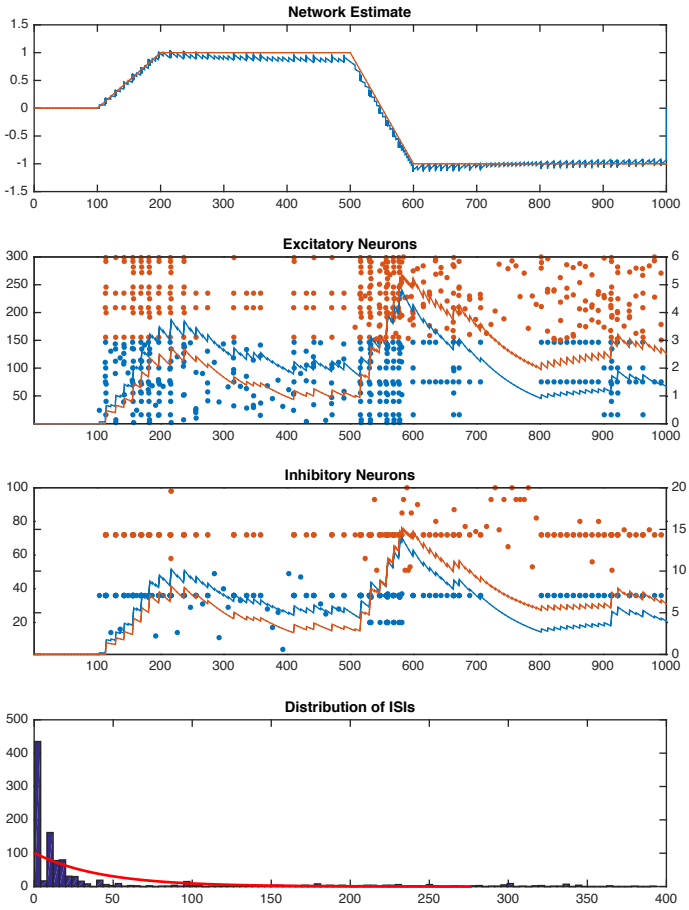

FIGURE 4. Performance of our model in tracking a square wave function (see text for details). First three panels as in figure 3. Bottom panel is a histogram of interspike intervals (ms), collapsing across all neurons. Red line is best fitting exponential function. Tracking error = .0815

3.2. **Model behaviour.** Figures 4 and 5 show the performance of our model tracking two different one-dimensional variables. In figure 4, the network is performing perfect integration of a variable x with dynamics $\dot{x} = c(t)$ (i.e. $\mathbf{A} = 0$), where $c(t)$ is a positive square wave followed by a negative one:

$$c(t) = \begin{cases} 10 & .1 \leq t \leq .2 \\ -20 & .5 \leq t \leq .6 \\ 0 & else \end{cases}$$

In figure 5, the network tracks a variable x with dynamics $\dot{x} = -5x + c(t)$, effectively performing leaky integration over the input commands $c(t)$, which consist of

a positive and a negative pulse corrupted by Gaussian white noise $\eta \sim \mathcal{N}(0, 10)$:

$$c(t) = \begin{cases} 10 + \eta & .1 \leq t \leq .2 \\ -20 + \eta & .5 \leq t \leq .6 \\ 0 & else \end{cases}$$

Comparing these figures to the simulation results of Boerlin *et al* (2013) (particularly figure S3), it is easy to see that our model behaves in a qualitatively similar fashion. Firstly, the network's tracking performance is excellent, with relative error of .0815 and .0665 in the perfect and leaky integrator cases, respectively. Secondly, spike trains are Poisson-like and irregular, with exponential-like distribution of ISI's. Lastly, firing rates are within a plausible 1-3Hz range.

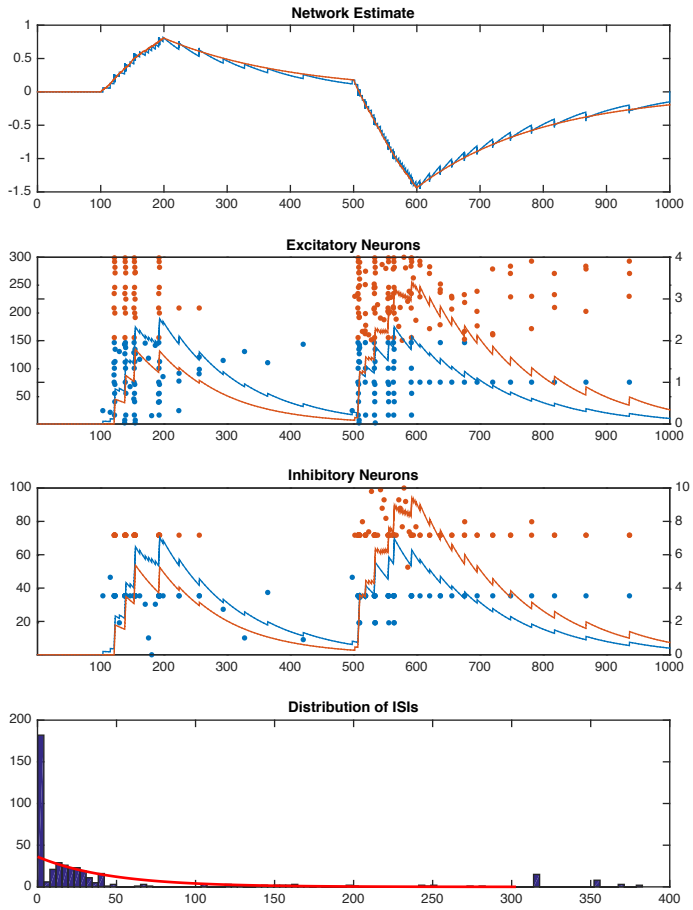

FIGURE 5. Performance of our model in tracking a decaying signal with two Gaussian pulses injected into the system (see text for details). Plots exactly as in figure 4. Tracking error = .0665

As stressed above, an important characteristic of this class of spiking networks is that there is a particularly tight balance between excitation and inhibition. [5] We find this as well with our model. In the top panels of figures 6 and 7, we see that the firing rates of the inhibitory and excitatory populations, while operating at different levels (because of the differing population sizes), fluctuate in a perfectly synchronized fashion: every time an excitatoy neuron spikes, an inhibitory neuron spikes. Indeed a strong correlation between firing rates was found, with Pearson $r > 0.99$ for both populations in each case. The histogram in the second panel, explains this: almost every excitatory spike is followed by an inhibitory spike in the next timestep, i.e. 0.1ms later. Another particularly salient feature of the class of spiking networks we are considering is that subthreshold membrane potentials are correlated between neurons with similar decoding kernels (bottom panels, blue line), and anticorrelated between neurons with different decoding kernels (red line). [5]

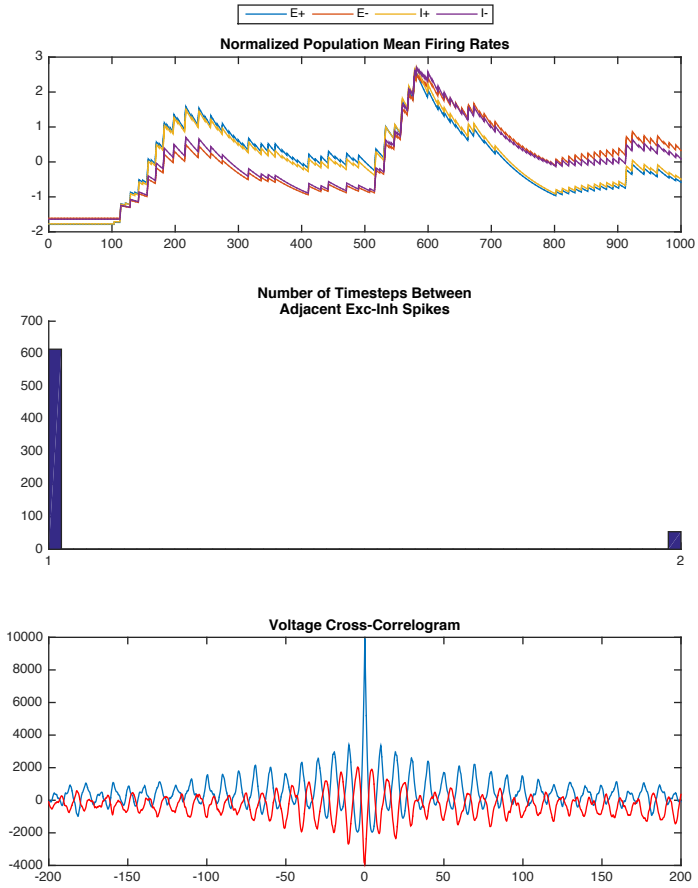

FIGURE 6. E/I balance of the integrator network. Top panel shows population mean firing rates for each subpopulation, converted to z-scores to facilitate comparison. Note that fluctuations in the firing rates of excitatory and inhibitory neurons with corresponding decoding kernels are correlated at a very fine timescale, Pearson $r = .995, 992, p < .0001$ for positive kernel and negative kernel populations, respectively. Middle panel shows histogram the number of timesteps between an excitatory spike and the next closest inhibitory spike (from any neuron in the network). Bottom panel shows the cross-correlation between the membrane potentials of two excitatory neurons with similar kernels (blue) and with opposite kernels (red).

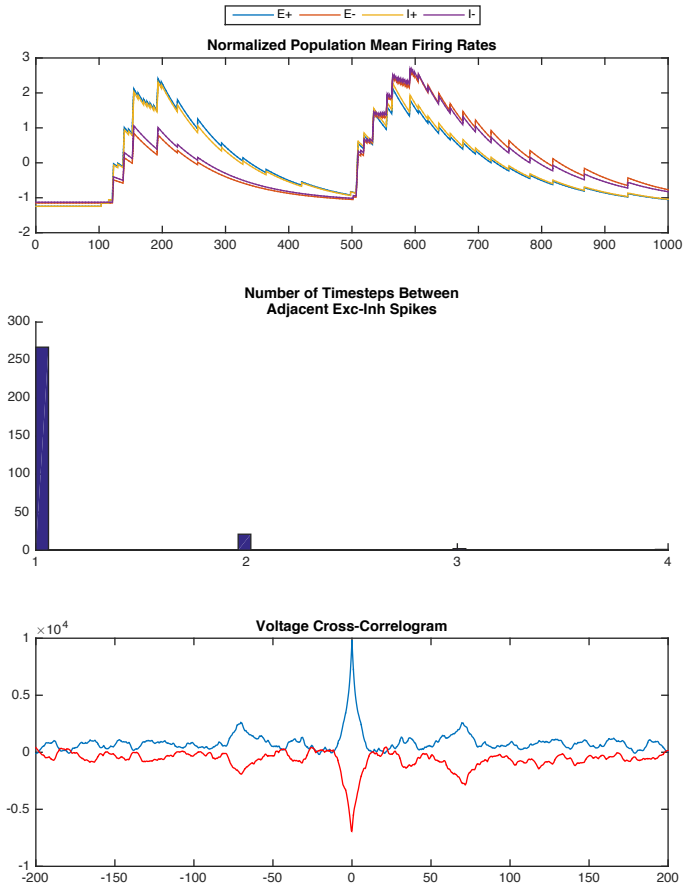

FIGURE 7. E/I balance of the leaky integrator network. Plots exactly as in figure 6. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, Pearson $r = .997, .994, p < .0001$.

We ask whether such

3.3. **Synaptic weights.** One modification to the model that might loosen the correlation between excitatory and inhibitory currents is weakening the synaptic weights between excitatory and inhibitory neurons. This would decrease the strength of EPSPs in inhibitory neurons, thus requiring more more excitatory spikes to trigger an action potential. To test this, we took the same network simulated above and multiplied all the excitatory decoding kernels by 0.1. This has the effect of decreasing all synaptic weights (except for the recurrent within-population inhibitory synapses) and lowering the spiking threshold for excitatory neurons, while leaving the inhibitory population thresholds untouched.

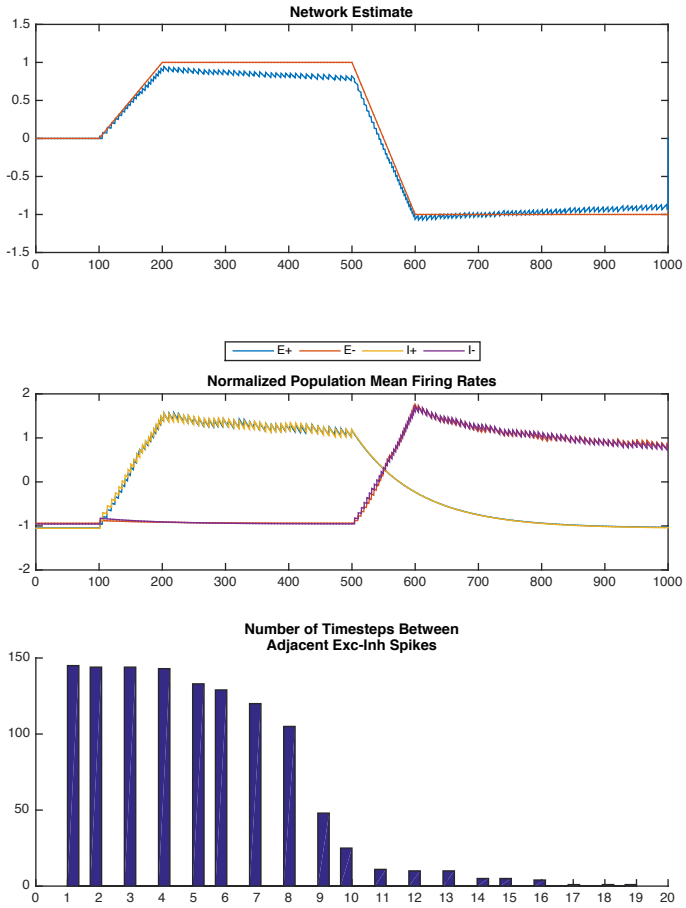

FIGURE 8. Tracking performance and E/I balance of the network with weakened excitatory synapses, see text for details. See previous figures for detailed description of each plot. Tracking error = .116. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, $r = 1.00, .999, p < .0001$.

Our results are shown in figure 8. The network's tracking performance remains high, while the E/I balance is slightly altered: rather than an inhibitory spike immediately following every excitatory spike, it is often the case that 1-2 ms transpire after an excitatory spikes prior to an inhibitory spike being elicited. However, the correlation in firing rates was in fact slightly strengthened.

Another approach is to depart further from the mathematically derived model by injecting some noise into the synaptic strengths, thus deviating them from their theoretically optimal values. This additionally leads to a more biologically plausible network by eliminating the perfect symmetry in synaptic connection strengths between excitatory and inhibitory neurons and between excitatory neurons with opposite kernels, while keeping them correlated. [20] We multiplied every synaptic

weight by a rectified Gaussian variable with mean 1 and standard deviation 0.1. We found, however, that performance suffers and the tight excitatory/inhibitory (E/I) balance remains (figure 9), even though again the time between adjacent excitatory and inhibitory spikes was spread over a much larger range.

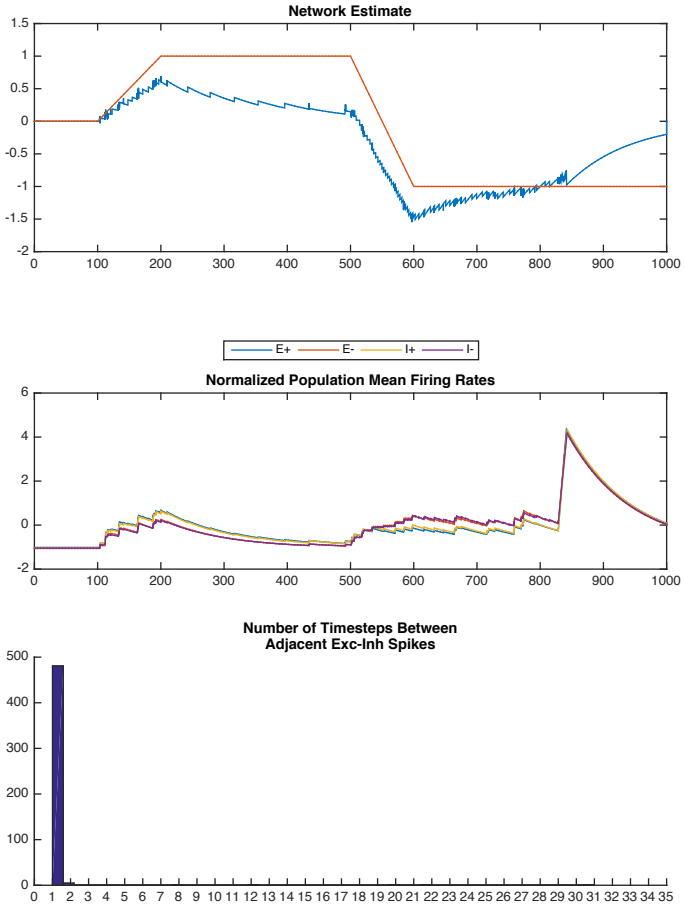

FIGURE 9. Tracking performance and E/I balance of the network with noisy excitatory synapses, see text for details. Tracking error = .580. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, $r = .998, .999, p < .0001$.

3.4. **Sparse connections.** Another move towards biological plausibility worth exploring is altering the connectivity of the network. Previous work has shown that connectivity in the cortex is quite sparse, with a rate of connectivity of around 11%. [20] On the other hand, inhibitory-excitatory connectivity seems to be quite dense, one experiment finding 70% of randomly sampled interneuron-pyramidal cell pairs to be connected. [7]

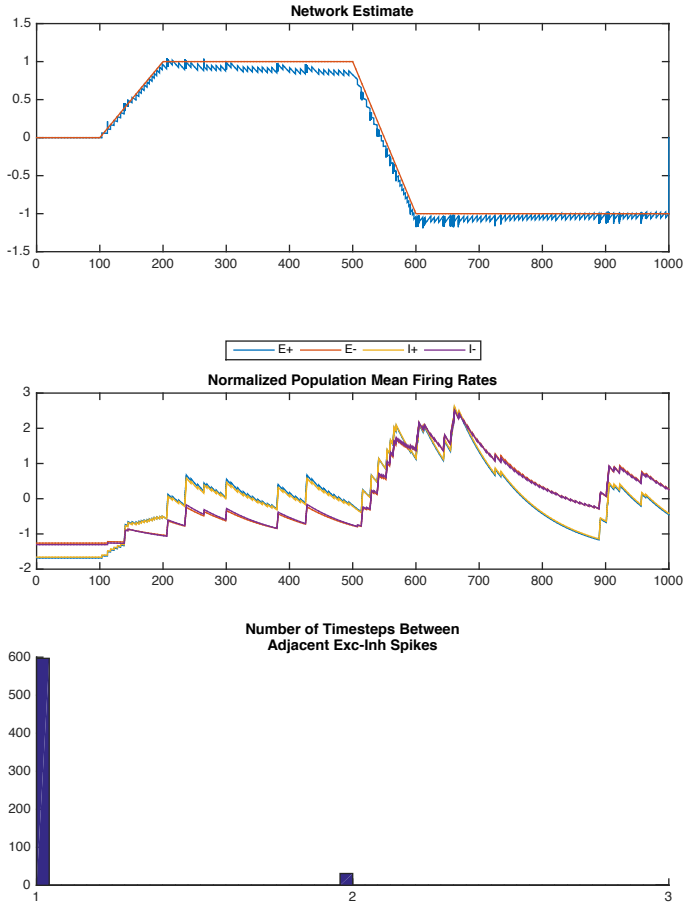

FIGURE 10. Tracking performance and E/I balance of the network with sparse connections, connectivity rate = 0.11. Tracking error = .0901. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, $r = .999, .999, p < .0001$.

We tested each of these scenarios by appropriately adjusting the decoding kernels. In the fully sparse network case, the kernels were adjusted so that the proportion of non-zero synaptic weights between any two populations was 0.11.[3] In the dense inhibitory connectivity case, the same excitatory kernels were used and new decoding kernels were drawn for the inhibitory neurons, enforcing them to all

[3]To maintain comparability between models, we used exactly the same decoding kernels as in all previous simulations, modifying them by setting a random sample of their components to 0 to achieve the desired connection rates. All non-zero synaptic weights this case are thus the same as their homologues in the above tested models.

be non-zero. This resulted in connection rates of 0.11 between excitatory neurons and 0.33 between excitatory and inhibitory neurons. [4]

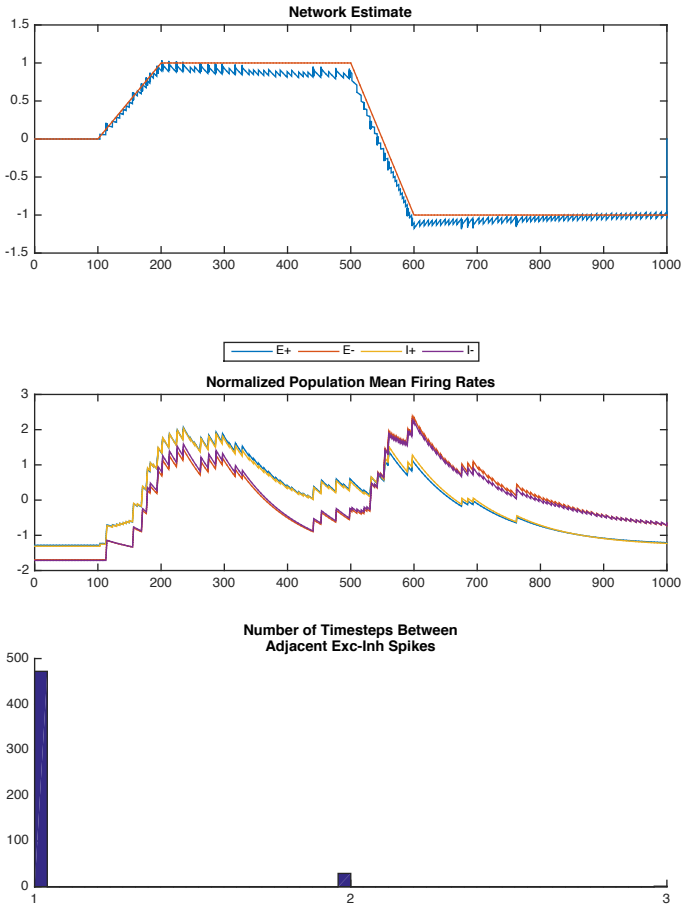

FIGURE 11. Tracking performance and E/I balance of the network with sparse excitatory connections and dense inhibitory connections. The connectivity is set to 0.11 for excitatory-excitatory connections and 0.33 for excitatory-inhibitory connections, see text for details. Tracking error = .106. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, $r = .999, .997, p < .0001$.

The results for the fully sparse and partially sparse networks are shown in figures 10 and 11, respectively. Tracking performance remains high, and the tight correlation in excitation and inhibition remains.

[4]Because of the interdependency of synaptic weights in our network (arising from the fact that they are derived from the individual neuron decoding kernels), it is impossible to obtain a connectivity rate of .11 between excitatory neurons and .7 between excitatory and inhibitory neurons as is suggested to be the case by the above cited evidence.

3.5. **Decoder leak rates.** An alternative construction we have not considered in our derivation of the network dynamics is that the inhibitory and excitatory estimates have different representational leak rates λ_d. This is of particular note because if we assume that this leak rate is slower for inhibitory neurons than for excitatory neurons, then the derivation results in a new set of inhibitory "slow" synapses between inhibitory neurons (see appendix E for the resulting equation). These recurrent connections could thus potentially lead to inhibition of the inhibitory population, disrupting the E/I balance.

However, we found that only very small differences in these leak rates keep the tracking performance within a suitable range. Figure 12 shows the results of simulations of the model when $\lambda_d = 11$ for the excitatory population and $\lambda_d = 10$ for the inhibitory population. Tracking performance suffers greatly and the E/I balance remains unchanged.

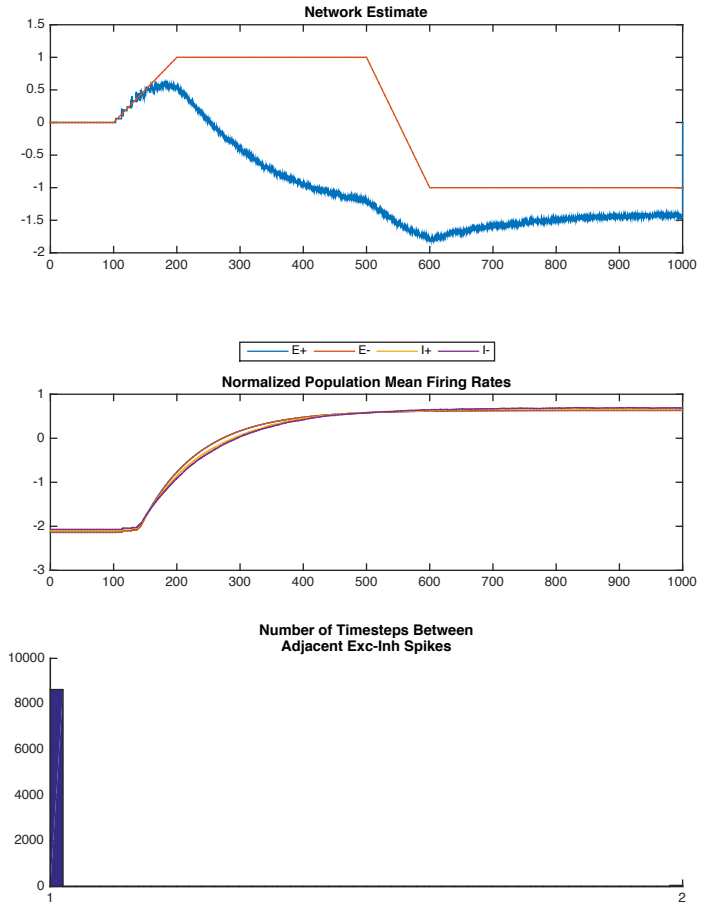

FIGURE 12. Tracking performance and E/I balance of the network with different representational leak rates for the excitatory and inhibitory population, leading to a new set of inhibitory "slow" synapses between inhibitory neurons. Here, $\lambda_d^{\mathrm{E}} = 11$ and $\lambda_d^{\mathrm{I}} = 10$. Tracking error = 1.251. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, $r = .999, .997, p < .0001$.

3.6. **Firing costs.** As noted in appendix B, our final dynamical equations implemented (approximately) in our model are derived from minimizing a cost function including the error function of equation 3 as well as a sum over individual neuron firing rates. In other words, the network minimizes a cost function composed of prediction error plus firing rates, the latter component weighted by a linear firing cost parameter v. In the above simulations, this parameter was set to 0 for the inhibitory neurons. Figure 13 shows the simulation results from a model incorporating a firing cost for inhibitory neurons, which has the effect of raising the spiking threshold. Naturally, it now takes on average more excitatory spikes to make an inhibitory neuron spike, reflected by the larger spread of time elapsed

between adjacent excitatory-inhibitory spikes in our simulation (figure 13, bottom panel). However, corresponding inhibitory and excitatory population mean firing rates remained highly correlated.

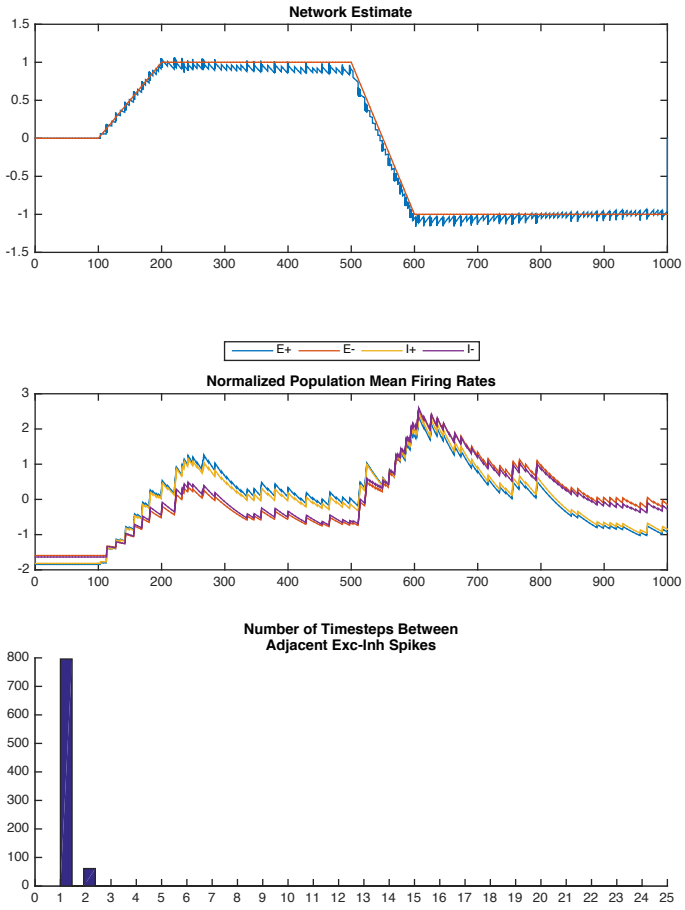

FIGURE 13. Tracking performance and E/I balance of the network incorporating a linear firing cost for inhibitory neurons, here set to 10^{-5}. Tracking error = .076. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, $r = .995, .996, p < .0001$.

3.7. **Slow synapses.** Finally, we tried incorporating biophysically realistic synapses by setting the synaptic kernel to a scaled decaying exponential, $\kappa(t) = Ze^{-\lambda_r t}$. This results in "fast" synapses having an instant rise time to Z, followed by a decay with rate λ_r. By convolving this kernel with the decaying exponential kernel h_d to get the "slow" syanpses, "slow" post-synaptic potentials have the form of a difference-of-exponentials $Z'(e^{-\lambda_d t} - e^{-\lambda_r t})$ with finite rise time $1/\lambda_r$ and decay rate λ_d ($\lambda_r > \lambda_d, Z' > 0$). This model also included a background noise term $\eta \sim \mathcal{N}(0, 0.1)$ added to the membrane potentials.

We found that this model performed very poorly, not being able to track the variable at all (figure 14). This is surprising given that previous efforts to similarly modify the unrealistic instant synapses of the original model have succeeded. [3, 15] It is thus worth noting here that our model stands apart from two previous approaches in that (1) it generealizes to any dynamical system $\mathbf{Ax}$ (as opposed to ref. [3], where they only track a one-dimensional signal with A=$-\lambda_d$), and (2) it obeys Dale's law (as opposed to ref. [15], where they incorporate Hodgkin-Huxley-type ionic currents that then need to be compensated for by recurrent connections that complicate the translation of the network to one that obeys Dale's law). Due to time constraints we were not able to analyze our model to understand why it performed so poorly. However, it suggests that might not be possible to construct a spiking network derived as above that simultaneously includes realistic synaptic dynamics and obeys Dale's law. Our simulations also suggest that incorporating such synapses only strengthens the correlations between excitatory and inhibitory signals.

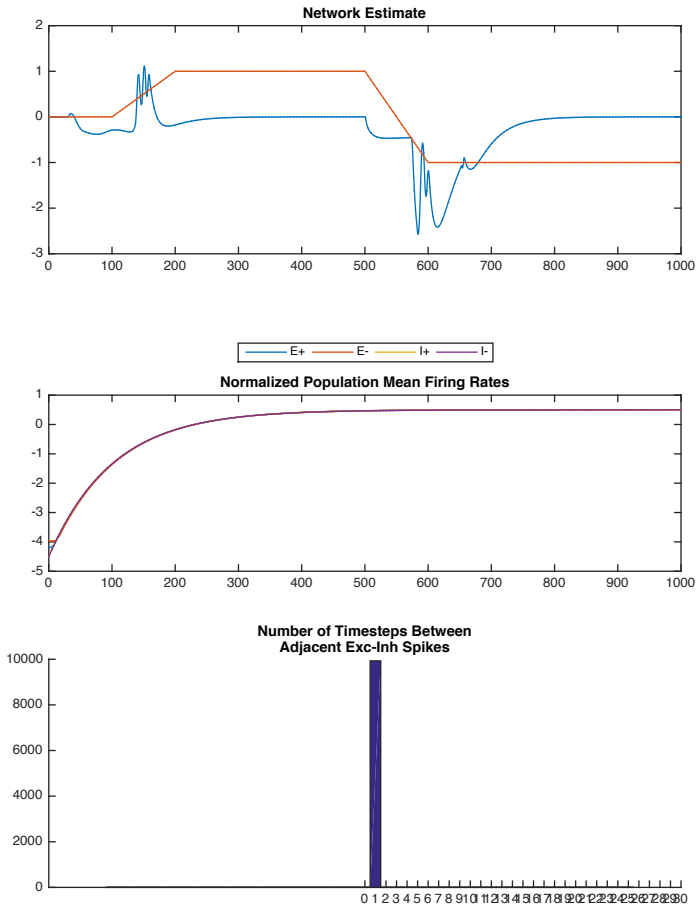

FIGURE 14. Tracking performance and E/I balance of the network incorporating biophysically realistic synapses (see text for details). Tracking error = .99. Excitatory-inhibitory firing rates significantly correlated for neurons with decoding kernels in the same direction, $r = 1.0, .999, p < .0001$.

4. A related rate-network approach

It is worth noting the relation between the model considered here and a related rate-network designed to maintain the representation of a static variable while the neurons interact dynamically. [6] The so-called FEVER network is constructed by a simple rule derived from the individual neuron dynamics and the dynamics of the variable being represented (in this case equal to 0, since it is static). I proceed to show that the same logic can be exploited to derive a rate network that can track an arbitrary J-dimensional signal with linear dynamics and inputs $c(t)$, as the spiking network we have been considering does.

Because the model in question is a rate network rather than a spiking network, individual neuron dynamics are defined over their firing rates, rather than their

membrane potentials:

$$\frac{dr_i}{dt} = -\lambda_F r_i + \sum_{j=1}^{N} \mathbf{W}_{ij} r_j + \sum_{k=1}^{J} \mathbf{\Omega}_{ik} c_k \tag{11}$$

where $\mathbf{W}_{ij}$ is the weight of the "synaptic" connection from neuron j to neuron i and $\mathbf{\Omega}_{ik}$ is the weight of the contribution of the kth external input signal to the ith neuron. We then assume the network's representation $\hat{\mathbf{x}}(t)$ is a linear sum over each neuron's decoding kernels, as above, except that the decoding kernels are now weighted by each neuron's firing rate:

$$\hat{\mathbf{x}}(t) = \mathbf{\Gamma}\mathbf{r} \tag{12}$$

where $\mathbf{\Gamma}$ is a $J \times N$ matrix in which the ith column corresponds to the decoding kernel of the ith neuron, for tracking a J-dimensional variable (and $\mathbf{r}$ is a $N \times 1$ vector with the firing rates for each neuron).

Given these two assumptions, we can derive an equation for our desired synaptic weights $\mathbf{W}$ by equating the derivative of our estimate do the dynamics of the variable $\mathbf{x}$ we wish to track:

$$\frac{d\hat{\mathbf{x}}}{dt} = \mathbf{\Gamma}\frac{d\mathbf{r}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{c} \tag{13}$$

Plugging in equation 11:

$$\mathbf{\Gamma}(-\lambda_F \mathbf{r} + \mathbf{W}\mathbf{r} + \mathbf{\Omega}\mathbf{c}) = \mathbf{A}\mathbf{x} + \mathbf{c} \tag{14}$$

Substituting in the network estimate $\hat{\mathbf{x}}$ for $\mathbf{x}$, plugging in equation 12, and rearranging, we get

$$(\mathbf{\Gamma}\mathbf{W} - \lambda_F \mathbf{\Gamma} - \mathbf{A}\mathbf{\Gamma})\mathbf{r} = (\mathbf{I} - \mathbf{\Gamma}\mathbf{\Omega})\mathbf{c} \tag{15}$$

To get a solvable equation for the synaptic weights, we assume that $\mathbf{\Omega} = \mathbf{\Gamma}^T(\mathbf{\Gamma}\mathbf{\Gamma}^T)^{-1}$. In other words, we assume that $\mathbf{\Gamma}$ has a right pseudo-inverse and that $\mathbf{\Omega}$ is it. This results in the right side of equation 15 reducing to $\mathbf{0}$, such that we can now obtain $\mathbf{W}$ by solving the following equation:

$$\mathbf{\Gamma}\mathbf{W} - \lambda_F \mathbf{\Gamma} - \mathbf{A}\mathbf{\Gamma} = 0 \tag{16}$$

Relating this to the original FEVER rule, we can express it as a constraint on the network decoding kernels, where every neuron's decoding kernel $\mathbf{\Gamma}_i$ must satisfy the following rule:

$$\mathbf{\Gamma}_i = \sum_k \mathbf{\Gamma}_{ki} \mathbf{A}_k - \sum_j \mathbf{W}_{ji} \mathbf{\Gamma}_j \tag{17}$$

where the single subscripts on matrices index columns. In other words, a given neuron's decoding kernel is a sum of two components: (1) a sum over the decoding kernels of each of its post-synaptic neurons, weighted by the synaptic connections, and (2) a sum over the coefficient vectors for each component of the J-dimensional

dynamical system being tracked, weighted by the decoding kernel elements corresponding to each of these components. In the spirit of Druckmann *et al*, we title this equation the DynoFEVER rule.

We can show that solutions exist for equation 16. We start by considering the set of eigenvectors $\vec{v}_m$ and eigenvalues λ_m of $\mathbf{W}^T$, and assuming that each row of the decoding kernel matrix is a weighted sum of these eigenvectors:

$$\mathbf{\Gamma}_i = \sum_m a_m^{(i)} \vec{v}_m \tag{18}$$

assuming the convention henceforth that $\mathbf{\Gamma}_i$ is the transpose of the ith row of $\mathbf{\Gamma}$ (i.e. an $N \times 1$ matrix). Rearranging equation 16, we have that $\mathbf{\Gamma W} = \lambda_F \mathbf{\Gamma} + \mathbf{A\Gamma}$. Solving for $\mathbf{\Gamma}_i$, this translates to:

$$\mathbf{W}^T \mathbf{\Gamma}_i = \lambda_F \mathbf{\Gamma}_i + \sum_j \mathbf{A}_{ij} \mathbf{\Gamma}_j \tag{19}$$

Since by definition $\mathbf{W}^T \vec{v}_m = \lambda_m \vec{v}_m$, equation 18 gives us

$$\mathbf{W}^T \mathbf{\Gamma}_i = \sum_m a_m^{(i)} \lambda_m \vec{v}_m \tag{20}$$

Plugging equations 18 and 20 into equation 19,

$$\sum_m a_m^{(i)} \lambda_m \vec{v}_m = \lambda_F \sum_m a_m^{(i)} \vec{v}_m + \sum_j \mathbf{A}_{ij} \sum_m a_m^{(j)} \vec{v}_m \tag{21}$$

Rearranging, we get

$$\sum_m \vec{v}_m (\lambda_m - 1) a_m^{(i)} = \sum_m \vec{v}_m \sum_j \mathbf{A}_{ij} a_m^{(j)} \tag{22}$$

resulting in the following equation:

$$(\lambda_m - 1) a_m^{(i)} = \sum_j \mathbf{A}_{ij} a_m^{(j)} \tag{23}$$

Solutions to this equation, along with equation 18, yields both $\mathbf{\Gamma}$ and $\mathbf{W}$, providing the necessary components to build the DynoFEVER network.

By construction, this network's estimate will evolve over time exactly as $\mathbf{x}(t)$. The case is indeed quite similar to the spiking network considered above, which is also derived from the dynamics of $\mathbf{x}(t)$. Is there a formal relationship here? At some level there has to be, since the dynamics of their respective estimates are, by construction, closely matched.

Relating their individual components, however, turns out to be non-trivial. In both cases the dynamics of the tracked variable are built into the network, but in completely different ways. The DynoFEVER rule is derived by equating the dynamics of the network estimate to the dynamics of the tracked variable. In contrast, the dynamics of the spiking network estimate are defined *a priori* (equation 2). Rather than fitting the rate or estimate dynamics to the tracked variable, we

constructed the spiking network by imposing an optimal spiking rule for individual neurons and then deriving the membrane potential dynamics from this rule. Thus, the discrepancy in construction between the two models arises at (at least) two levels. Firstly, the dynamics of $\mathbf{x}$ are built into the network in completely different ways: equation 3 vs. 13. Secondly, the resulting dynamical equations define the behaviour of the network at separate functional levels: firing rates vs. membrane potentials.

While rate networks are not biologically plausible in themselves, they can be exploited to construct spiking networks that perform the same computations[1], so the question of how these two models relate is of clear relevance to the challenge of building functional spiking networks. It is worth noting here that the DynoFEVER network can, in principle, be made to obey Dale's law. [6] The question then becomes what the E/I balance would look like in a spiking network derived from the DynoFEVER network, and whether it might in fact be formally equivalent to the spiking network described above. We leave this an open question for further investigation.

5. Discussion

In summary, we have constructed a spiking network crudely more biologically plausible than that put forth by Boerlin *et al* (2013). We did not succeed in approaching the biophysical relevance of more biologically-oriented models derived in the same fashion [15], but our model did obey broader neurophysiological principles (e.g. it obeyed Dale's law, as opposed to the model in [15]). We showed that, like the idealized mathematical model on which it is based, our model requires a tight balance of excitation and inhibition, which is maintained across a wide variety of parameter settings. Even altering the synaptic strengths and spiking thresholds did not change this.

Is such tight E/I balance empirically supported? While it is well known that excitatory and inhibitory currents to a cortical neuron tend to be balanced on average[17, 13, 25, 9], evidence for correlation at such a fine temporal scale [24, 8] is not as strong. Furthermore, strong temporal correlations between neuron membrane potentials like those observed here are not typically observed in biological neurons, although at least one study has found such correlations in primary visual cortex. [26]

Empirically, the picture is not at all clear. From a theoretical perspective, our results support the hypothesis that functional spiking networks require a particularly tight E/I balance [2, 5]. We modified the original balanced spiking network of Boerlin *et al* in a number of ways to make it more biologically plausible and found that under all these variations the tight correlation in excitatory and inhibitory firing rates remained.

Finally, we stressed two theoretical questions that remain to be answered. Firstly, the possibility of incorporating realistic synaptic into our model holds much promise

for constructing biologically realistic functional spiking networks. Even more powerful would be to find a way to get the biophysically motivated model of Schwemmer *et al* (2015) to obey Dale's law. Secondly, the lack of empirical evidence for the "tight" E/I balance in these networks poses the question of whether they can be constructed in a different manner. We have proposed we try working from the DynoFEVER network to deriving a spiking network.

References

[1] Abbott, L., DePasquale, B., and Memmesheimer, R.-M. Building functional networks of spiking model neurons. *Nature Neuroscience 19*, 3 (2016), 350–355.

[2] Boerlin, M., Machens, C. K., and Denève, S. Predictive coding of dynamical variables in balanced spiking networks. *PLoS Comput Biol 9*, 11 (2013), e1003258.

[3] Chalk, M., Gutkin, B., and Deneve, S. Neural oscillations as a signature of efficient coding in the presence of synaptic delays. *bioRxiv* (2015), 034736.

[4] Dayan, P., and Abbott, L. F. *Theoretical neuroscience*, vol. 806. Cambridge, MA: MIT Press, 2001.

[5] Denève, S., and Machens, C. K. Efficient codes and balanced networks. *Nature Neuroscience 19*, 3 (2016), 375–382.

[6] Druckmann, S., and Chklovskii, D. B. Neuronal circuits underlying persistent representations despite time varying activity. *Current Biology 22*, 22 (2012), 2095–2103.

[7] Fino, E., and Yuste, R. Dense inhibitory connectivity in neocortex. *Neuron 69*, 6 (2011), 1188–1203.

[8] Gentet, L. J., Avermann, M., Matyas, F., Staiger, J. F., and Petersen, C. C. Membrane potential dynamics of gabaergic neurons in the barrel cortex of behaving mice. *Neuron 65*, 3 (2010), 422–435.

[9] Haider, B., Duque, A., Hasenstaub, A. R., and McCormick, D. A. Neocortical network activity in vivo is generated through a dynamic balance of excitation and inhibition. *The Journal of neuroscience 26*, 17 (2006), 4535–4545.

[10] Koch, C. *Biophysics of computation: information processing in single neurons*. Oxford university press, 2004.

[11] London, M., Roth, A., Beeren, L., Häusser, M., and Latham, P. E. Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature 466*, 7302 (2010), 123–127.

[12] Mainen, Z. F., and Sejnowski, T. J. Reliability of spike timing in neocortical neurons. *Science 268*, 5216 (1995), 1503–1506.

[13] Okun, M., and Lampl, I. Instantaneous correlation of excitation and inhibition during ongoing and sensory-evoked activities. *Nature neuroscience 11*, 5 (2008), 535–537.

[14] Rumelhart, D. E., McClelland, J. L., Group, P. R., et al. *Parallel distributed processing*, vol. 1. IEEE, 1988.

[15] Schwemmer, M. A., Fairhall, A. L., Denéve, S., and Shea-Brown, E. T. Constructing precisely computing networks with biophysical spiking neurons. *The Journal of Neuroscience 35*, 28 (2015), 10112–10134.

[16] Shadlen, M. N., and Newsome, W. T. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *The Journal of neuroscience 18*, 10 (1998), 3870–3896.

[17] Shu, Y., Hasenstaub, A., and McCormick, D. A. Turning on and off recurrent balanced cortical activity. *Nature 423*, 6937 (2003), 288–293.

[18] Smolensky, P., and Legendre, G. *The harmonic mind: From neural computation to optimality-theoretic grammar (Vol. 1: Cognitive architecture).* MIT Press, 2006.
[19] Softky, W. R., and Koch, C. The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps. *The Journal of Neuroscience 13*, 1 (1993), 334–350.
[20] Song, S., Sjöström, P. J., Reigl, M., Nelson, S., and Chklovskii, D. B. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biol 3*, 3 (2005), e68.
[21] Thalmeier, D., Uhlmann, M., Kappen, H. J., and Memmesheimer, R.-M. Learning universal computations with spikes. *arXiv preprint arXiv:1505.07866* (2015).
[22] Tolhurst, D. J., Movshon, J. A., and Dean, A. F. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision research 23*, 8 (1983), 775–785.
[23] Vreeswijk, C. v., and Sompolinsky, H. Chaotic balanced state in a model of cortical circuits. *Neural computation 10*, 6 (1998), 1321–1371.
[24] Wehr, M., and Zador, A. M. Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex. *Nature 426*, 6965 (2003), 442–446.
[25] Xue, M., Atallah, B. V., and Scanziani, M. Equalizing excitation-inhibition ratios across visual cortical neurons. *Nature 511*, 7511 (2014), 596–600.
[26] Yu, J., and Ferster, D. Membrane potential synchrony in primary visual cortex during sensory stimulation. *Neuron 68*, 6 (2010), 1187–1201.

6. Appendices

6.1. **A. Modifying the spiking network to obey Dale's law.** One outstanding detail of the network derived in section 2 is that any given neuron may have outgoing excitatory and inhibitory connections, thus violating Dale's law. This can be fixed by partitioning the network into two subpopulations of excitatory and inhibitory neurons, respectively. The excitatory neurons receive feed-forward external input $\mathbf{c}(t)$ and track $\mathbf{x}(t)$ as above, with the inhibitory connections replaced by synapses from inhibitory neurons that track the excitatory population estimate $\hat{\mathbf{x}}^{\mathrm{E}}$ and take as input spikes from the excitatory population.

We thus derive the dynamics of the inhibitory neurons as we did for the non-Dale's law neurons above, but now minimizing the error function with respect to the excitatory estimate $\hat{\mathbf{x}}^{\mathrm{E}}$, rather than $\mathbf{x}$:

$$E^{\mathrm{I}}(t) = \int_0^t d\tau \left(\hat{\mathbf{x}}^{\mathrm{E}}(\tau) - \hat{\mathbf{x}}^{\mathrm{I}}(\tau)\right)^2 \tag{24}$$

where I indexes the inhibitory population and E indexes the excitatory population. By minimizing $E^{\mathrm{I}}(t)$ over spike times, expressing the resulting spiking condition as a membrane potential and a spiking threshold, and taking the derivative of the equation for membrane potential (and adding a leak term), we get the following dynamics for inhibitory neurons:

$$\frac{d\mathbf{V}^{\mathrm{I}}}{dt} = -\lambda_V \mathbf{V}^{\mathrm{I}} + {\mathbf{\Gamma}^{\mathrm{I}}}^T \mathbf{\Gamma}^{\mathrm{E}} \mathbf{o}^{\mathrm{E}} * \kappa - {\mathbf{\Gamma}^{\mathrm{I}}}^T \mathbf{\Gamma}^{\mathrm{I}} \mathbf{o}^{\mathrm{E}} * \kappa \tag{25}$$

with spiking thresholds $T_i^{\mathrm{I}} = \|\mathbf{\Gamma}_i^{\mathrm{I}}\|^2\kappa(\epsilon)/2$, where $\mathbf{\Gamma}^{\mathrm{I}}$ is the matrix of decoding kernels of the inhibitory neurons, and $\mathbf{\Gamma}^{\mathrm{E}}$ its excitatory counterpart. Note that inhibitory neurons only take input from excitatory neurons, and that all synapses to and from these neurons are of the "fast" variety. Importantly, all synaptic inputs from the excitatory population E are excitatory and all synaptic inputs from I are inhibitory, thus obeying Dale's law. It should be noted here that this in fact only holds if the components of $\mathbf{\Gamma}^{\mathrm{E}}$ and $\mathbf{\Gamma}^{\mathrm{I}}$ are all positive or all negative. We will return to this point below.

Having seen that the derivation of the non-Dale's law network leads to dynamics that ensure faithful representation of the tracked variable $\mathbf{x}(t)$, we can be sure that $\hat{\mathbf{x}}^{\mathrm{I}}(t)$ will be a good estimate of $\hat{\mathbf{x}}^{\mathrm{E}}(t)$. Thus we can now modify the excitatory neuron dynamics to obey Dale's law by replacing $\hat{\mathbf{x}} \equiv \hat{\mathbf{x}}^{\mathrm{E}}(t)$ in equation 6 with $\hat{\mathbf{x}}^{\mathrm{I}}(t)$, leading to

$$\frac{d\mathbf{V}^{\mathrm{E}}}{dt} = -\lambda_V \mathbf{V}^{\mathrm{E}} + \mathbf{\Gamma}^{\mathrm{E}T}\mathbf{c} + \mathbf{\Gamma}^{\mathrm{E}T}\mathbf{A}\mathbf{x} + \lambda_d \mathbf{\Gamma}^{\mathrm{E}T}\hat{\mathbf{x}}^I - \mathbf{\Gamma}^{\mathrm{E}T}\mathbf{\Gamma}^{\mathrm{I}}\mathbf{o}^{\mathrm{I}} * \kappa \tag{26}$$

Again exploiting the fact that $\mathbf{x} \approx \hat{\mathbf{x}}^E$ and $\hat{\mathbf{x}}^I \approx \hat{\mathbf{x}}^E = \mathbf{\Gamma}^{\mathrm{E}}\mathbf{o}^{\mathrm{E}} * \kappa * h_d$, we obtain dynamics that obey Dale's law:

$$\frac{d\mathbf{V}^{\mathrm{E}}}{dt} = -\lambda_V \mathbf{V}^{\mathrm{E}} + \mathbf{\Gamma}^{\mathrm{E}T}\mathbf{c} + \mathbf{\Gamma}^{\mathrm{E}T}(\mathbf{A} + \lambda_d \mathbf{I})\mathbf{\Gamma}^{\mathrm{E}}\mathbf{o}^{\mathrm{E}} * \kappa * h_d - \mathbf{\Gamma}^{\mathrm{E}T}\mathbf{\Gamma}^{\mathrm{I}}\mathbf{o}^{\mathrm{I}} * \kappa \tag{27}$$

Here, excitation is mediated by recurrent connections $\mathbf{\Gamma}^{\mathrm{E}T}(\mathbf{A} + \lambda_d \mathbf{I})\mathbf{\Gamma}^{\mathrm{E}}$ dependent on the dynamics of the variable being tracked, and inhibition (as well membrane potential resets after spiking) is implemented via excitatory-inhibitory synaptic weights equal to the dot product of the decoding kernels of the pre-synaptic and post-synaptic neurons (see figure 1).

A few caveats to this architecture have been ignored. Firstly, if the dynamics of the tracked variable are negative and faster than the dynamics of the network estimate (e.g. $\mathrm{eig}(\mathbf{A}) < -\lambda_d$) then $\mathbf{A} + \lambda_d$ will yield negative recurrent connection weights between excitatory neurons. This can be easily accommodated by using the inhibitory population estimate $\hat{\mathbf{x}}^{\mathrm{I}}(t)$ instead of $\hat{\mathbf{x}}^{\mathrm{E}}(t)$ in the third term of equation 27, as in equation 26. Secondly, as was already mentioned briefly, the excitatory/inhibitory status of each of the synapses we have constructed is contingent on the decoding weights of the excitatory and inhibitory populations being all positive or all negative, such that $\mathbf{\Gamma}^{\mathrm{E}T}\mathbf{\Gamma}^{\mathrm{I}} > 0$. This constrains the network estimate $\hat{\mathbf{x}}^E(t)$ to being able to actively decrease or increase but not both, depending on whether its decoding kernels are all negative or all positive, respectively. This can be easily fixed by adding another population of excitatory and inhibitory neurons with decoding kernels of an opposite sign. By similarly following the same derivation structure as above, the connections between these populations can be defined such that the network as a whole can behave (in theory) exactly like the non-Dale's law case above (see ref. [2] supplementary materials for the full derivation).

Figure 1 provides a schematic of this network architecture, under the assumption that $(\mathbf{A} + \lambda_d) > 0$.

6.2. **B. Incorporation of firing costs.** We note that in our derivation of the spiking network we omitted the incorporation of linear firing costs. Such costs are necessary to obtain biologically realistic irregular spike trains. The corresponding dynamics are obtained by adding a linear firing cost to the error function that we minimize over:

$$E(t) = \int_0^t d\tau (\mathbf{x}(\tau) - \hat{\mathbf{x}}(\tau))^2 + v|\mathbf{r}(\tau)| \tag{28}$$

where $\mathbf{r}$ is a vector of the firing rates of all neurons and $|.|$ is the L1 norm. This simply results in an increase of the neuron thresholds by $v\lambda_d$: $T_i = \frac{1}{2}(\|\mathbf{\Gamma}_i^{\mathrm{I}}\|^2\kappa(\epsilon) + v\lambda_d)$.

6.3. **C. Simulation parameters.** Except where noted, the following parameter values were used: Note that the decoding kernels were drawn randomly only once.

Parameter	**Value**	**Parameter meaning**
N^{E}	150 neurons	number of excitatory neurons
N^{I}	50 neurons	number of inhibitory neurons
$\Phi_{ij}^{\mathrm{E}}, \Phi_{ij}^{\mathrm{I}}$	$\sim$ Binom$(1, 0.7)$Unif$(.06, .1)$	decoding kernels of excitatory and inhibitory neurons with positive kernels
$\Psi_{ij}^{\mathrm{E}}, \Psi_{ij}^{\mathrm{I}}$	$\sim$ Binom$(1, 0.7)$Unif$(-.1, -.06)$	decoding kernels of excitatory and inhibitory neurons with negative kernels
dt	10^{-4} sec.	Euler timestep for numerical integration
λ_d	10Hz	representational (decoder) leak rate
λ_V	5Hz	voltage leak rate
$v^{\mathrm{E}}, v^{\mathrm{I}}$	$10^{-5}, 0$	linear firing costs for excitatory and inhibitory neurons

Except where noted, the same decoding kernels were used for all simulations.

Unless otherwise noted (e.g. the leaky integrator network), all simulations were constructed to track variable $\mathbf{x}$ with dynamics $\dot{\mathbf{x}} = \mathbf{c}(t)$, with $\mathbf{c}(t)$ defined as for the integrator network (square wave function).

6.4. **D. Quantifying network performance: relative error.** Tracking error was quantified by computing the relative error between the network estimate $\hat{\mathbf{x}}$

and the tracked variable $\mathbf{x}$:

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|} \tag{29}$$

6.5. **E. Equations for a network with different excitatory and inhibitory representational leak rates λ_d.** Going through the same derivation as in appendix A, but now with different leak rates for the excitatory population λ_d^{E} and the inhibitory population λ_d^{I}, we get the following altered dynamics for inhibitory neurons:

$$\frac{d\mathbf{V}^{\mathrm{I}}}{dt} = -\lambda_V \mathbf{V}^{\mathrm{I}} + (\lambda_d^{\mathrm{I}} - \lambda_d^{\mathrm{E}}){\mathbf{\Gamma}^{\mathrm{I}}}^T \mathbf{\Gamma}^{\mathrm{I}} \mathbf{o}^{\mathrm{E}} * \kappa * h_d + {\mathbf{\Gamma}^{\mathrm{I}}}^T \mathbf{\Gamma}^{\mathrm{E}} \mathbf{o}^{\mathrm{E}} * \kappa - {\mathbf{\Gamma}^{\mathrm{I}}}^T \mathbf{\Gamma}^{\mathrm{I}} \mathbf{o}^{\mathrm{E}} * \kappa \tag{30}$$

As long as $\lambda_d^{\mathrm{E}} > \lambda_d^{\mathrm{I}}$, this will result in the addition of "slow" recurrent synapses between inhibitory neurons (the second term in the equation), on top of the "fast" ones (the last term).