Behavioral/Systems/Cognitive

Higher-Dimensional Neurons Explain the Tuning and Dynamics of Working Memory Cells

Ray Singh[1] **and Chris Eliasmith**[1,2]
Departments of [1]Systems Design Engineering and [2]Philosophy, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Measurements of neural activity in working memory during a somatosensory discrimination task show that the content of working memory is not only stimulus dependent but also strongly time varying. We present a biologically plausible neural model that reproduces the wide variety of characteristic responses observed in those experiments. Central to our model is a heterogeneous ensemble of two-dimensional neurons that are hypothesized to simultaneously encode two distinct stimuli dimensions. We demonstrate that the spiking activity of each neuron in the population can be understood as the result of a two-dimensional state space trajectory projected onto the tuning curve of the neuron. The wide variety of observed responses is thus a natural consequence of a population of neurons with a diverse set of preferred stimulus vectors and response functions in this two-dimensional space. In addition, we propose a taxonomy of network topologies that will generate the two-dimensional trajectory necessary to exploit this population. We conclude by proposing some experimental indicators to help distinguish among these possibilities.

Key words: **computational model; neural dynamics; population coding; reward uncertainty; working memory; cognitive**

Introduction

The majority of work related to working memory takes stably persistent activity to be an indicator that a cell is participating in remembering a stimulus (Fuster, 1973; Gnadt and Andersen, 1988; Funahashi et al., 1989; Zhang, 1999; Taube and Basset, 2003). However, recent experiments have shown that the majority of the neurons in areas associated with working memory have dynamically varying activity during the delay period.

Using macaques trained to perform a vibrotactile discrimination task, Romo et al. (1999) have captured this dynamic activity of individual neurons in the prefrontal cortex (PFC). Their experiment requires the comparison of two mechanical vibrations (f1 and f2) separated by 3–6 s. The task has been analyzed as consisting of three distinct phases: (1) the loading phase in which f1 is registered; (2) the storage phase in which f1 is maintained; and (3) the decision phase in which f1 and f2 are compared. Spiking activities of individual neurons are recorded throughout these time periods. In this study, we are primarily concerned with the mechanisms of memorization and thus focus on the activity during the storage phase.

The neural responses (see Fig. 2*a–f*) have been nominally categorized by their monotonic relationship to the base stimulus, f1 (Romo et al., 1999). Activities that increase with f1 are "positive monotonic," and those that decrease with f1 are "negative monotonic." Surprisingly, the activities of most neurons are not persistent but display a characteristic ramping up or down behavior. Consequently, the responses are further distinguished by periods of monotonicity. Neurons that are monotonic throughout the delay period are deemed "persistent," whereas those that are only monotonic during the beginning or end of the delay are designated "early" and "late" neurons, respectively.

In this modeling study, we propose a very simple means of capturing the wide variety of observed responses in a neurally plausible network. The current understanding of working memory is that such areas realize a simple one-dimensional line attractor or a set of such attractors (Wang, 1999; Seung et al., 2000; Brody et al., 2003; Miller et al., 2003). Unfortunately, this assumption makes it very difficult to capture the wide variety of responses observed by Romo et al. (1999). This was recently demonstrated by the model of Miller et al. (2003), in which they were unable to capture responses like those in shown in Figure 2*c* with a fairly complex, six-population model. In addition, past models have difficulty incorporating the wide diversity of neural response functions observed in these areas, usually assuming that any heterogeneity of responses is a result of the "messiness" of neural systems. We demonstrate that by considering a two-dimensional model, all of the categories of responses observed by Romo et al. (1999) can be captured. We show that this is possible only because we explicitly incorporate the heterogeneity observed in neural systems into the model. As a result, both the messiness of the system and the higher-dimensional sensitivity of neurons play an important role in explaining the experimental data in a simple, one-population network.

Materials and Methods

To generate and exploit this two-dimensional population, we follow the methodology of Eliasmith and Anderson (2003). Briefly, we begin by randomly choosing neural parameters that fall within biologically plau-

Received Nov. 11, 2005; revised Jan. 23, 2006; accepted Feb. 23, 2006.

This work was supported by grants from the Natural Science and Engineering Research Council (Canada) (261453-03), Canadian Foundation for Innovation (3358401), and Ontario Innovati (3358501). We thank J. Conklin, B. Tripp, and P. Miller for valuable discussions.

Correspondence should be addressed to Dr. Chris Eliasmith at the above address. E-mail: celiasmith@uwaterloo.ca.

DOI:10.1523/JNEUROSCI.4864-05.2006

 0270-6474/06/263667-12$15.00/0

sible regimens. We then suggest a plausible population-level encoding and decoding relationship with the relevant stimuli that defines the two-dimensional encoding. Finally, we determine how to instantiate higher-level dynamics (i.e., an appropriate state space trajectory) using this population of neurons. The resulting single-cell behavior is then compared with observed data to determine whether the posited encoding, decoding, and dynamics are reasonable. Here, we describe each of these steps in detail, with comparisons to more familiar applications to one-dimensional representations.

Central to our results is a general characterization of representation as neural encoding and decoding. The population of neurons in the model is a heterogeneous collection of adapting leaky integrate-and-fire (LIF) neurons. As described in the study by Eliasmith and Anderson (2003), more complex single-neuron models can be used as well, but there is a significant computational cost without much gain in realism, especially in the context of this particular model. The current present in the soma of a particular neuron can be described generically as follows:

$$J_i(\mathbf{x}) = \alpha_i \langle \tilde{\phi}_i \cdot \mathbf{x} \rangle + J_i^{bias} + \eta_i, \quad (1)$$

where $Ji(\mathbf{x})$ is the input current to neuron i, $\mathbf{x}$ is the vector variable of the stimulus space encoded by the neuron, α_i is a gain factor, $\tilde{\phi}_i$ is the preferred-direction vector of the neuron in the stimulus space, J_i^{bias} is a bias current that accounts for background activity, and η_i models neural noise. Notably, the dot product, $\langle \tilde{\phi}_i \cdot \mathbf{x} \rangle$, describes the relationship between a high-dimensional physical quantity (e.g., a stimulus) and the resulting scalar signal describing the input current.

For clarity, it is worth comparing the responses of one- and two-dimensional neurons. In the one-dimensional case, the preferred-direction vector is either $+1$ or -1. Neurons with a preferred direction that is positive are "on" neurons. That is, they will increase their firing rate as the value of the stimulus variable increases. The opposite is true for the negative, or "off," neurons. The addition of a second dimension generalizes this characterization such that the preferred directions now lie at any direction on the unit circle, rather than just ± 1. Thus, as a constant magnitude stimulus vector sweeps past the preferred-direction vector of the neuron, the firing rate of the neuron will trace out a typical cosine-type tuning curve, with peak firing at the preferred direction. As the magnitude of the stimulus increases at a constant direction (e.g., the preferred direction), the firing rate of the neuron will increase proportionally, just as it did in the one-dimensional case (see Fig. 3). In both cases, these tuning curves are, in fact, the result of both Equation 1 and a neural nonlinearity.

In particular, in our model, the time course of the somatic voltage in response to this current evolves as a standard LIF neuron, with the addition of adaptation. These dynamics are captured by (Koch, 1999) as follows:

$$dV_i/dt = -(V_i(1 + RG_{adapt}) - J_i(\mathbf{x})R)/\tau_i^{RC}$$

$$dG_{adapt}/dt = -G_{adapt}/\tau_{adapt}, \quad (2)$$

where V_i is the somatic voltage, R is the leak resistance, τ_i^{RC} is the RC time constant, and G_{adapt} is the time-varying conductance modulated by τ_{adapt}. The system is integrated until the membrane potential, V_i, crosses the neuron threshold, V_{th}, at which point a $\delta(t - t_{in})$ spike is generated, G_{adapt} is increased by G_{inc}, and V_i is reset to zero for the duration of the refractory period, τ_i^{ref}. The inclusion of adaptation helps account for the observed effects of stimulus onset (see Fig. 2, gray bars). Notably, including adaptation does not adversely affect the derivation or overall behavior of the model.

As mentioned, it is important for the model to include the heterogeneity typical of single-cell responses observed in the cortex. Using Equations 1 and 2 as a model of neuron behavior, we randomly select a set of neural parameters. In particular, the preferred-direction vectors, $\tilde{\phi}_i$, are drawn from a uniform distribution around the two-dimensional unit circle. The distribution is uniform primarily because we have no indication that it should be otherwise, and this distribution has been shown appropriate for other cortical models (Georgopoulos et al., 1984). The gain and bias current, α_i and J_i^{bias}, are chosen such that the maximum firing rates are randomly assigned to neurons but evenly distributed between 20 and 100 Hz, to match the data of Romo et al. (1999). The RC time constant is chosen to lie in $\tau^{RC} = 5$–15 ms, typical membrane time constant values, and the adaptation constant is set to lie in $\tau_{adapt} = 1$–200 ms to reflect the wide variety of adaptation in pyramidal neurons. The refractory period is set to $\tau_i^{ref} = 1$ ms, again a typical value, and $G_{inc} = 20$ nS, which has been shown to effectively match cortical adaptation (Koch, 1999). In addition, during the simulation, independent Gaussian noise, $\eta_i = N(0, 0.1)$, is injected into the soma to account for various sources of neural noise (e.g., spike jitter, thermal fluctuations, neurotransmitter variations, etc.). In summary, given available evidence, the model population was closely matched to the parameter regimens that describe the kind of cortical population we suspect Romo et al. (1999) encountered during their recordings. Because many of the parameters are statistically matched, the precise responses of the model population will vary between runs.

We have now completed our characterization of neural encoding (Eqs. 1 and 2). Next we must address how these neurons can use the information that they have encoded about the stimuli of interest. That is, we must define neural decoding to determine (1) what information regarding the stimulus has been encoded and (2) how the information can be used, transformed, or computed over in a neural circuit.

For each neuron in a neural population, we find a neural decoder, ϕ_i. This decoder is a least-squares optimal weight that can be applied to the neural activities for estimating the encoded information in the population (see Appendix, Neural representation, for methods used to determine decoders). Together, the elements of Figure 1 show the steps involved in encoding and decoding a square wave and a ramp input in a one-dimensional neural population. Specifically, Figure 1*b* shows the spikes that result from encoding an input signal. Because we have sorted these neurons, and removed adaptation, the encoded information is easily evident in the spike pattern. Figure 1*c* depicts example postsynaptic currents (PSCs) that would result in a subsequent population that receives these spikes. This depicts temporally decoded (or filtered) spike trains, an example of $a_i(t)$ in Equation 3. Figure 1*a* demonstrates the results of summing over the population of neurons with such currents and weighting them by their decoders (black line). That is, it shows a decoded estimate $\hat{x}(t)$ of the original signal $x(t)$. So, the difference between the input and output signals in Figure 1*a* indicates how well this population has encoded the information in the original signal. To transform this encoded information (i.e., to compute some function of the input), the same methods can be used to find decoders for each such transformation [see Eliasmith and Anderson (2003) for detailed discussion]. This understanding of neural representation generalizes to the two-dimensional case. Rather than scalar weights, the decoders are two-dimensional vectors, but the methods do not otherwise change.

To this point, we have discussed the two-dimensional representation used in our model. It is this characterization of representation that explains why we are able to produce the variety of results observed in the neural system (see Results). This is because it is the path that the network takes through this representational space that provides an explanation of the data. However, we are also interested in understanding how this path itself is generated. To do so, we need to understand how network-level dynamics can be understood in the context of such representations.

After Eliasmith and Anderson (2003), we assume that the dynamics of the population can be expressed in terms of the dynamics of the signal(s) it is representing. As discussed in detail in the Appendix (see Neural dynamics), taking the neural representation to be the state variable of a dynamic system described by control theory leads to a general method for constructing complex, dynamic neural models. For instance, Eliasmith (2005) provides a comprehensive account of controlled spiking attractor networks (i.e., point, line, ring, plane, cyclic, and chaotic attractor networks) using these methods.

As discussed later, a number of possible dynamic systems can account for the behavior of the working memory neurons of interest here. However, to get a sense of how this variety of dynamics is used to construct a model, let us consider the simple example of a one-dimensional integrator. This recurrent network has previously been well characterized (Seung, 1996; Seung et al., 2000; Koulakov et al., 2002; Eliasmith and

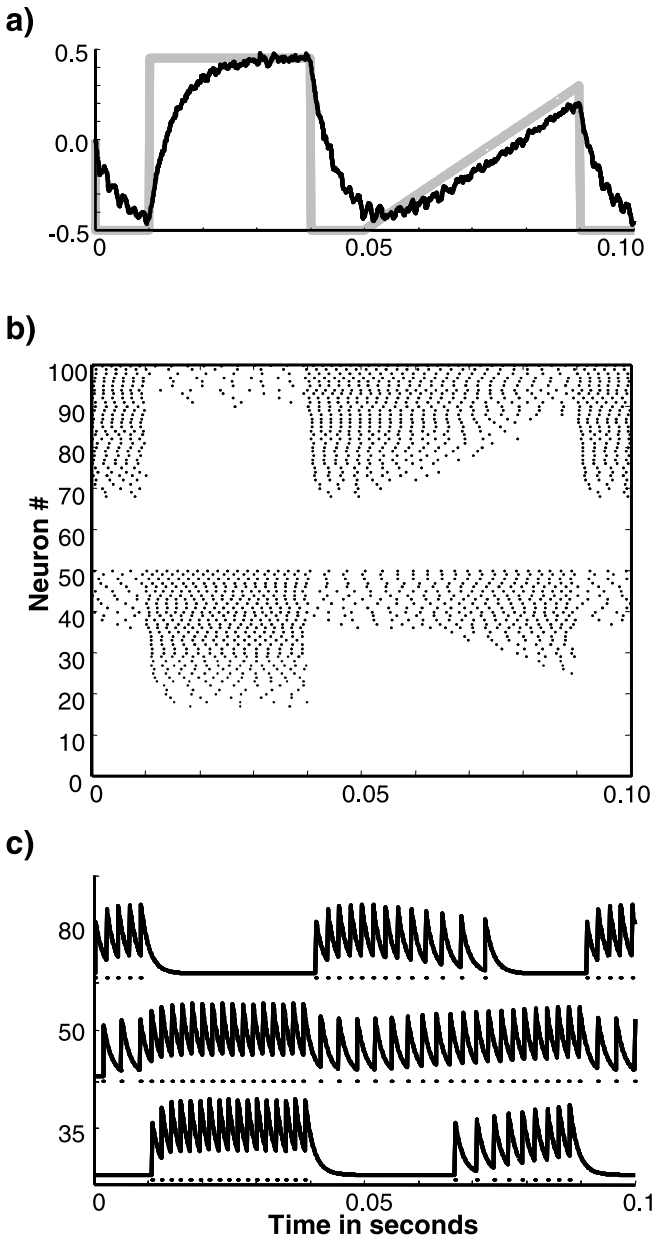

Figure 1. Population encoding and decoding of a square pulse and ramp signal. ***a***, The input $x(t)$ (gray line) and the decoded estimate, $\hat{x}(t)$ (black line), using a population of 100 one-dimensional LIF neurons. The weighted sum all of the PSCs yields the decoded estimate (black line). ***b***, The spike raster produced by encoding the input. Neurons are separated at $i = 50$ into on and off neurons ($\tilde{\phi}_i = +1$ and -1, respectively) and sorted by firing onset. ***c***, The rasters of neurons $i = 80$, 50, and 35 are plotted with their resulting PSCs. Notably, these were chosen to hint at the ramping response of early, persistent, and late firing seen in the data of Romo et al. (1999).

Anderson, 2003; Goldman et al., 2003). In summary, to implement a one-dimensional integrator of the signal x, we use our previously defined neural representation of x to determine the appropriate recurrent connection weights. If we would like the population to respect $\dot{x} = 0$ [i.e., that there is no change in x over time without input (which defines an integrator)], we must ensure that the representation encoded from the neural inputs is the same as the representation decoded from those inputs.

Combining Equations 1 and 2, we can write the encoding at the inputs as follows:

$$a_i(t) = G_i[\alpha_i\langle\tilde{\phi}_i x(t)\rangle + J_i^{bias}], \tag{3}$$

where G_i represents the encoding defined by Equation 2. We can also write the decoding of this encoded information as a weighted sum (as captured by Fig. 1) as follows:

$$\hat{x}(t) = \sum_{j=1}^{N} a_j(t)\phi_j, \tag{4}$$

where $a_j(t)$ is the neural activity (i.e., the postsynaptic filtered spike trains from neuron i, as shown in Fig. 1*c*, although G_i in Eq. 3 should be convolved with the PSC filter but is left out for simplicity; see Appendix, Neural representation, for a full characterization), N is the number of neurons in the population, and $\hat{x}(t)$ is the optimal linear estimate of x using the decoders. It should be noted that the subscripts i and j range over the same population of neurons, because the connections are recurrent.

With these definitions, we can now construct a one-dimensional neural integrator by allowing $\hat{x}(t) = x(t)$, thus substituting Equation 4 for Equation 3, giving the following:

$$a_i(t) = G_i\left[\alpha_i\left\langle\tilde{\phi}_i\sum_{j=1}^{N} a_j(t)\phi_j\right\rangle + J_i^{bias}\right] = G_i\left[\sum_{j=1}^{N}\omega_{ij}a_j(t) + J_i^{bias}\right], \tag{5}$$

where $\omega_{ij} = \alpha_i\langle\tilde{\phi}_i\phi_j\rangle$. Equation 5 now defines a neural integrator in terms of recurrent connection weights. That is, the circuit with recurrent connections that are defined by this weight matrix will have dynamics that are steady (i.e., an unchanging representation) under no input. As described in the Appendix (see Neural dynamics), this simple derivation can be generalized to arbitrary dynamics and more complex circuits (e.g., with an input signal). As well, it can be generalized to higher dimensions. So, in the two-dimensional case, the preferred-direction (encoding) and decoding vectors replace the encoding and decoding scalars, but otherwise the derivation is the same (see Fig. 6 for plots of these weight structures).

In essence, this derivation is much like those of Seung et al. (2000) and others. They use similar least-squares methods to tune a one-dimensional integrator. However, there are two important differences between past derivations and the one presented here (detailed in the appendices). First, Seung et al. (2000) and others do not characterize the role of the encoders and decoders as we have done. Usually, both are assumed to be free variables for finding the weights. In contrast, we have taken the encoders to be inferable directly from experimentally observed tuning curves. As a result, it is less clear how to generalize past methods to higher dimensions. Here, however, the derivation is the same, regardless of the dimensionality of the tuning curves. Second, we have introduced a general dynamics matrix into our weight derivations, which results in a standard form for the weights of $\omega_{ij} = \alpha_i\langle\tilde{\phi}_i\mathbf{A}'\phi_j\rangle$ (see Appendix, Neural dynamics). This is not evident in the simple one-dimensional integrator case, because the $\mathbf{A}'$ matrix is equal to 1 and not explicitly included in past derivations. However, as discussed in detail in Results, this characterization allows us to construct a wide variety of complex dynamics in higher-dimensional spaces.

To simulate the data of Romo et al. (1999), circuits derived in this manner were modeled using the Neural Engineering Simulator, which is available as an open source (http://sourceforge.net/projects/nesim). To match the experimental setup of Romo et al. (1999), seven evenly spaced step inputs are used to simulate the base stimulus (f1). The stimulus lasts for 0.5 s, and the delay period runs for 3 s as in the original experiments. The spiking activity of each neuron is collected. Again following the method used by Romo et al. (1999), poststimulus time histogram (PSTH) plots are generated by convolving the spike trains with Gaussian kernels ($\sigma = 150$ ms during the delay period; $\sigma = 50$ ms elsewhere). For all simulations, the PSC time constant is 100 ms. A summary of the parameters used can be found in Table 1.

Results

Here, we describe our simulation results and situate them with respect to past attempts at modeling the observed working mem-

Table 1. Model parameters

	Symbol	Range	Description
1.	$\max G_i[J(\mathbf{x})]$	20–100 Hz	Maximum firing rate
2.	$G_i[J(\mathbf{x})] = 0$	−1 to 1	Normalized **x**-axis intercept
3.	J_i^{bias}	Satisfies 1 and 2	Bias current
4.	α_i	Satisfies 1 and 2	Gain factor
5.	$\tilde{\phi}_i$	$\|\tilde{\phi}_i\| = 1$	Preferred-direction (encoding) unit vector
6.	τ_i^{RC}	5–15 ms	RC time constant
7.	τ_{adapt}	1–200 ms	Adaptation time constant
8.	G_{inc}	20 nS	Adaptation conductance
9.	τ_i^{ref}	1 ms	Refractory period
10.	τ_{PSC}	100 ms	PSC time constant

ory effects. We then provide a taxonomy of network topologies that give rise to the dynamics needed to explain the data of Romo et al. (1999).

Related work

Much effort has been spent designing systems that maintain a persistent signal after stimulus presentation, the presumed purpose of working memory. However, this approach is somewhat at odds with the data, which suggests that the majority of working memory signals are not persistent. Hence, our purpose here is to describe a neural model that reproduces the dynamics of the data of Romo et al. (1999). We are less concerned with neural mechanisms for signal persistence (Seung, 1996; Koulakov et al., 2002; Goldman et al., 2003). Nevertheless, as described above, our solution is related to that of Seung et al. (2000), and we provide some discussion of the robustness of the solution to fine-tuning and noise. However, here we address the more immediate question: How can memory be reliably encoded in a time-varying signal and what explains the wide variety of dynamics observed in working memory?

Miller et al. (2003) have recently proposed a model that addresses this question directly, and in the context of the vibrotactile discrimination task. Their model consists of a large network of LIF neurons in a locally structured circuit. To avoid the fine-tuning problem that plagues some neural integrators [like that of Seung et al. (2000)], Miller et al. (2003) use a collection of bistable groups to create a network of multi-stable states (Koulakov et al., 2002). However, they conclude that non-finely tuned networks do not properly reflect the data because such networks result in highly discontinuous neuron responses. Additionally, they demonstrate that fine tuning is a reasonable alternative, which is also supported by our results below.

Of greater interest here is how they attempt to reproduce the wide variety of observed neural dynamics. To do so, they propose a network of three neural integrator populations that capture the characteristic ramping up, down, and tonic behaviors. The populations are assumed to be assembled together with suitable excitatory and inhibitory connections. Each population consists of two subnetworks: one that supports negative monotonicity and one that supports positive monotonicity. Each subnetwork consists of 12 neuronal groups of 500 neurons each (the bistable integrators). The resulting subnetworks have 6000 neurons each for a total of 36,000 in the entire network (although each population of 12,000 neurons is simulated independently).

Although their model can broadly simulate the categorized responses, it is unable to reproduce the wide variety of neural responses seen in data set of Romo et al. (1999). For instance, neurons in the study by Miller et al. (2003) exhibit either tonic or ramping curves but not variations of both, as in Figure 2, *c* and *C*, where the high-frequency responses are ramping but the low-frequency responses are tonic. This is because, like other past characterizations of working memory (Zipser et al., 1993; Camperi and Wang, 1998; Reutimann et al., 2004), Miller et al. (2003) tacitly assume that neural responses in their populations encode only one-dimensional signals. As a result, monotonicity with respect to frequency is rendered independent from time-varying dynamics. These features are then further subdivided: monotonicity is split into positive and negative monotonic neurons, and time-varying dynamics are split into ramping up or down tendencies. Unfortunately, this divide-and-conquer approach has two limitations. First, it unnecessarily complicates matters. For instance, there is no need to explicitly simulate two oppositely ramping responses; using a two-dimensional population with randomly distributed preferred directions in the two-dimensional space automatically provides neurons with oppositely directed tuning curves that naturally account for this kind of behavior. Second, it results in missing some of the observed properties of neural tuning. In addition to not reproducing Figure 2, *c* and *C*, separating the two dimensions will result in ramping neurons always starting from an initial (background) position and "fanning" outward, although the opposite (fanning inward) is seen in Figure 2, *a* and *A*.

Simulation results

The empirical responses shown in Figure 2*a–f* clearly portray neurons that are both time and stimulus dependent. These dependencies are independent only in the sense that frequency monotonicity does not dictate time-varying behavior. This does not entail that the representation of these dimensions needs to be independent, however. Thus, we propose a model that views neurons as simultaneously sensitive to both quantities, with those sensitivities evenly and randomly distributed in a two-dimensional space (see Materials and Methods). In particular, we assume that the two-dimensional space of interest has, as its dimensions, "time" (i.e., a representation of time; but see Discussion) and frequency. Mathematically, we denote the two quantities as the parameterized vector $\mathbf{x}(t) = [F(t), T(t)]$ (Fig. 3). Let us first consider the role of this representation in explaining the data of Romo et al. (1999).

Neurons representing this space will maximally fire to some "preferred stimulus," which defines a direction in the two-dimensional space as depicted in Figure 3. As a result, the observed gradations in sensitivity in a particular direction should be a reflection of the heterogeneity of intrinsic neural response curves (where such curves are understood as the responses found by direct current injection). So, a representation of the two-dimensional space that consists of randomly distributed preferred directions, along with a variety of intrinsic neural response curves, should correspond to the different kinds of tuning shown in Figure 2. Notably, these curves have been selectively chosen to match the experimental data. It would be more informative to compare the entire distribution of tuning curves to determine how typical such neuron classes are. However, because we were unable to examine the complete original data set, we cannot affect this comparison. The even distribution that we have assumed for tuning curves results in a wide variety of responses, many of which are intermediate between the classes shown in Figure 2. Other distributions would result in more "clustered" preferred-direction vectors and thus fewer and more typical cell classes. Given the general tendency to observe high heterogeneity in the cortex, we take the even distribution to be a reasonable assumption.

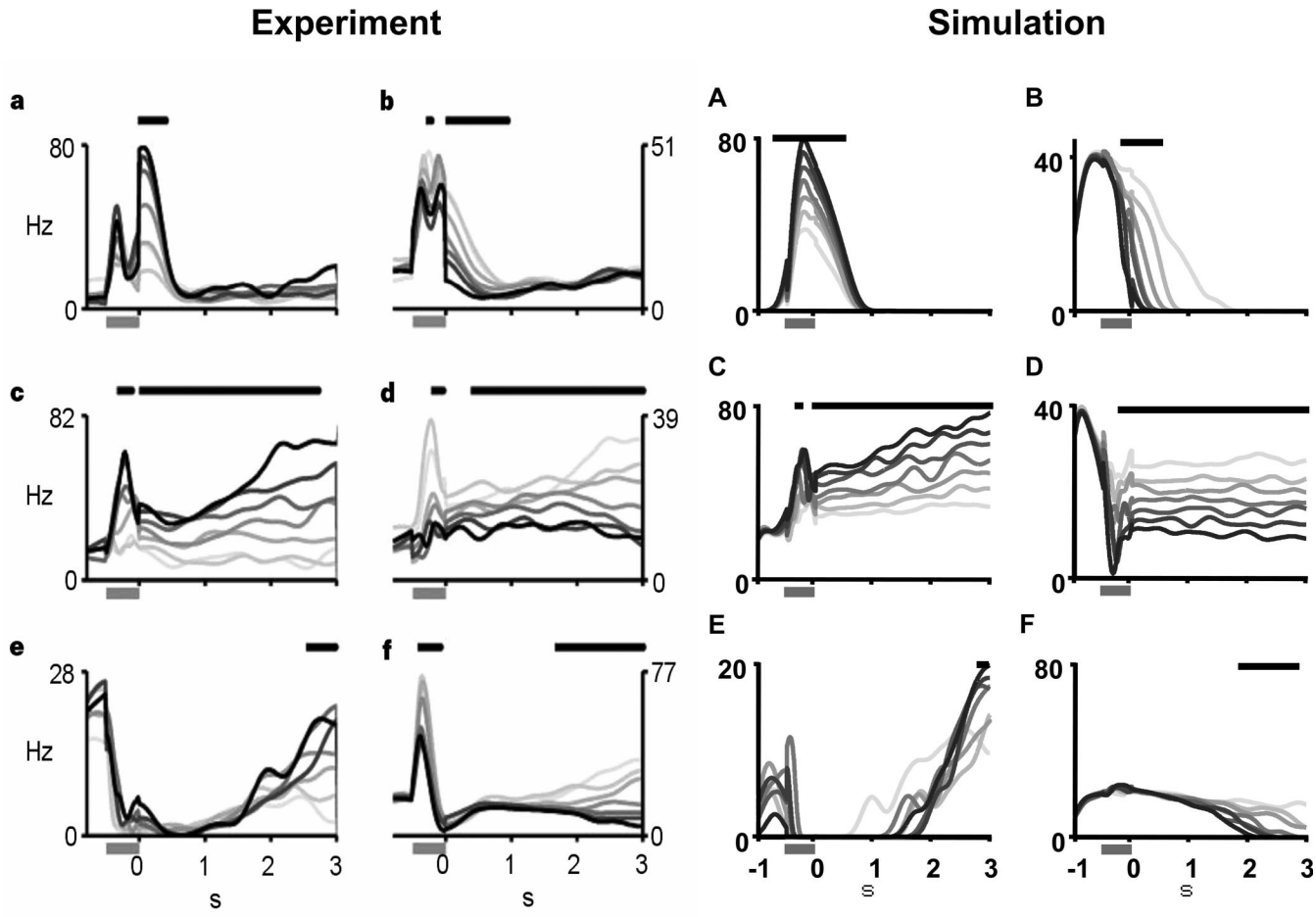

Figure 2. PSTH plots during memorization. The gray bars under the axes indicate the onset of the stimulus, and black bars above the graph mark periods of monotonicity. The higher stimulus frequency (f1) is marked with darker response curves. ***a***, ***c***, ***e***, Positive monotonic. ***b***, ***d***, ***f***, Negative monotonic. ***a***, ***b***, Early neurons. ***c***, ***d***, Persistent neurons. ***e***, ***f***, Late neurons. [Data are from Romo et al. (1999).] ***A–F***, Corresponding simulation results from the model shown.

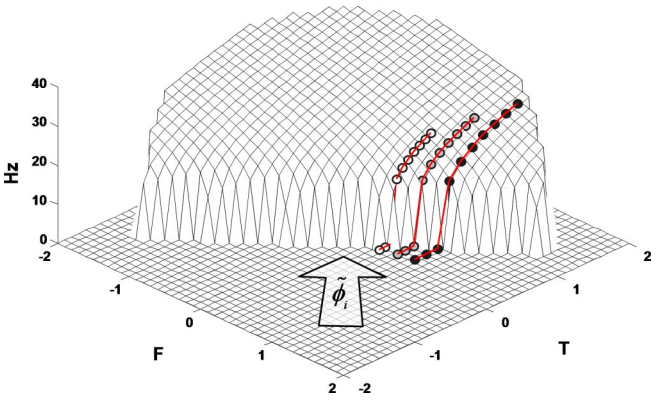

Figure 3. Tuning curve of a two-dimensional neuron defined by $G_i[\alpha_i\langle\tilde{\phi}_i \cdot X\rangle + J_i^{\text{bias}}$. For this neuron, the preferred-direction vector is $\tilde{\phi}_i = [-0.64, 0.766]$. Three state space trajectories, defined by $\mathbf{x}(t) = [F(t), T(t)]$, are shown projected onto the curve. For each trajectory, $F(t)$ is a constant, $T(t)$ is a ramping signal, and t ranges from 0 to 3 s. The units of $F(t)$ and $T(t)$ are open to interpretation, but one can view them as normalized frequency and time, respectively.

However, this characterization of the neural representation alone does not account for the changes over time of the neural responses. The specific values taken on by the two-dimensional quantity through time (i.e., the state space trajectory or dynamics) also play a significant role. Assuming a two-dimensional representation that consists of a time and frequency dimension, as we have, examination of the experiments of Romo et al. (1999) suggests that the observed dynamics result from a combination of a constant signal along the frequency dimension (the memory of f1) and a ramping signal in the time dimension.

To test this account, we built and simulated a network of two-dimensional neurons that realized this trajectory (see Materials and Methods). The results are plotted in Figure 2*A–F* and are juxtaposed with the experimental findings. The simulation reveals early, late, and persistent neurons that exhibit positive and negative monotonic responses, all of the classes of response described in the original data. So, the model demonstrates that the observed PSTHs can be understood as primarily the result of two-dimensional neural tuning curves and the dynamics of a two-dimensional quantity. Figure 4 provides a geometric explanation of how dynamics and tuning curves interact to result in the observed responses. In Figure 4*b*, filtered spike trains are shown as a function of the state space trajectory projected onto the two-dimensional tuning curve of a neuron. The path traveled on this surface is driven by the dynamics of the working memory signal. Each set of input signals produces a characteristic and systematic path across the surface. We can thus understand the observed variety of responses in the experiments: the PSTHs vary systematically with f1, yet generally maintain the same shape because there is a monotonically increasing time signal, $T(t)$, over a consistent (neural) nonlinearity.

The variety of observed tuning curves is thus explained by the

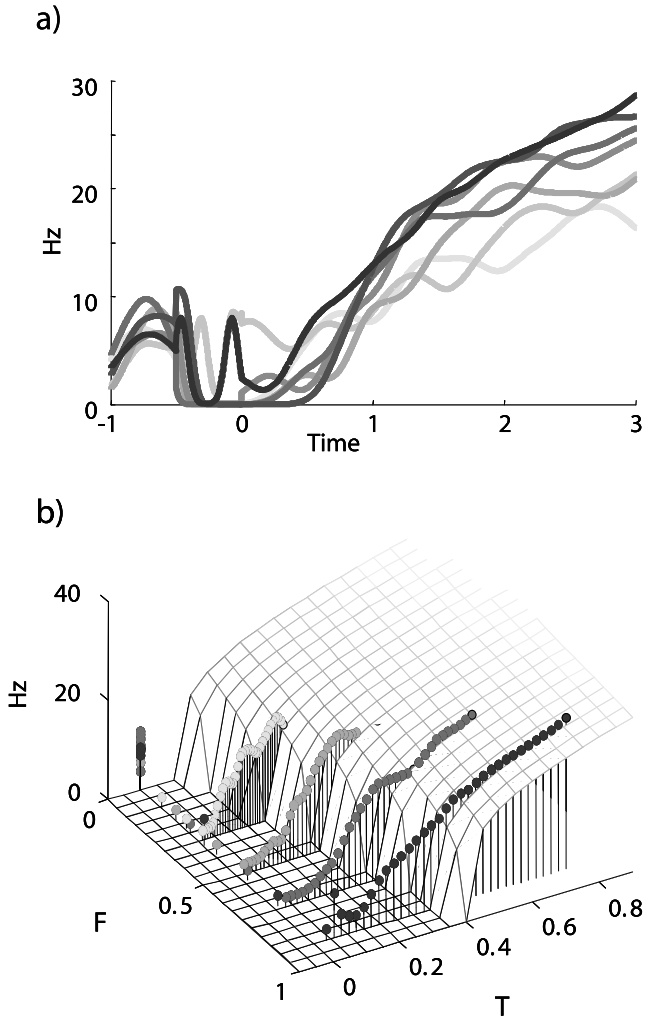

Figure 4. ***a***, Response curves of a late positive monotonic neuron. ***b***, The spiking rates can be seen as a mapping of the state space trajectory onto the two-dimensional tuning curve of the neuron. The trajectories of every other curve in ***a*** are shown. The responses do not lie exactly on the curve (as in Fig. 3) because of accumulated noise.

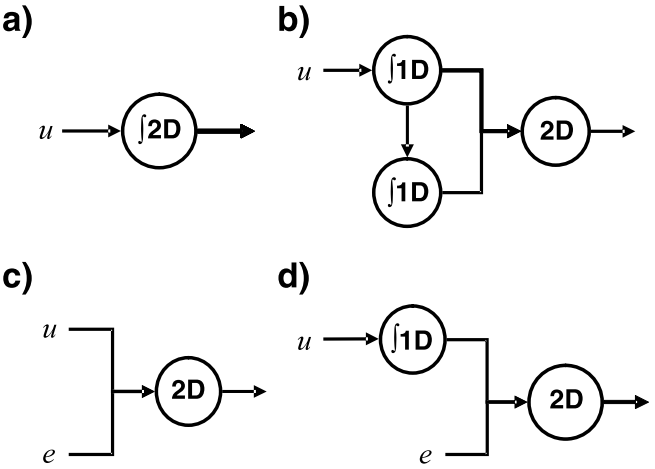

Figure 5. Network architectures that produce the observed responses. ***a***, A simple two-dimensional integrator. ***b***, Coupled one-dimensional integrators. ***c***, Projected external signals. ***d***, One-dimensional integrator with an external input. The circuits can be categorized as either using two-dimensional integration (***a***, ***c***) or two-dimensional projection (***b***, ***d***) with inputs that are locally generated (***a***, ***b***) or external (***c***, ***d***). 2D, Two-dimensional; 1D, one-dimensional.

distribution of preferred-direction vectors, $\tilde{\phi} = [\tilde{\phi}_f, \tilde{\phi}_t]$, the firing thresholds (along those vectors) in the two-dimensional space, and the neural nonlinearity. Generally, monotonicity is determined by $sign(\tilde{\phi}_f)$, and early, late, and persistent firing is characterized by $\tilde{\phi}_t$. As mentioned, we chose the vectors randomly from an even distribution over the unit circle and the response threshold randomly from an even distribution along the vector. Better fits to the likelihoods of observing various classes of responses could be made by altering these distributions to match the neural data (the complete data set was not available).

The key difference between this model and that of Miller et al. (2003) is in the representation of the content of memory. Rather than taking neurons to encode a series of one-dimensional signals, we understand them to encode a single two-dimensional space. As a result, the myriad of observed responses are a natural consequence of a heterogeneous population of individual neuron tuning curves directed in this higher-dimensional space. As a result, Figure 2, *c* and *C*, is observed in our model but not in that of Miller et al. (2003). Additionally, this approach results in a much more efficient use of neural resources. The results presented in Figure 2 are from a network of <3000 neurons (the organization of which is described below). This is an order of magnitude fewer neurons than in the model of Miller et al. (2003), despite a more complete characterization of the observed data, and the same degree of dynamic stability.

It is worth emphasizing that the neurons in our model are, biophysically speaking, no different than those in past models. The important difference is in how we have characterized what those biophysical states are used to represent. As a result, referring to these neurons as higher dimensional denotes the fact that they are sensitive to multiple physical dimensions concurrently. This, it should be noted, has nothing to do with the dimensionality of the equations used to describe the time course of the voltages and currents in the neuron model itself. What these results show, then, is that mapping biophysical states of cells into higher-dimensional representational spaces can more effectively explain the observed transitions between those biophysical states in real neural systems.

Dynamics

Our discussion so far leaves unresolved why or how this population has the particular trajectory through the state space that it does (i.e., a coupled ramp and constant). We address these questions in detail here. Systematically characterizing network-level dynamics is essential because the time-varying activity might be explained by a number of competing hypotheses. Some have suggested that the network encodes the passage of time, which could be used to deduce f1 (Brody et al., 2003; Reutimann et al., 2004). Others have posited that signals with the necessary time course are broadly projected to the PFC (Fiorillo et al., 2003). These latter signals are thought to encode reward uncertainty and could account for the observed dynamics in the PFC neurons. For the most part, our modeling effort is agnostic as to the exact nature of this quantity. However, here we discuss and model the architectural implications of these different kinds of explanations for the observed dynamics.

Recall that we have identified the necessary state space trajectory as including a constant signal related to f1 and a time-varying ramp signal, time. Clearly, then, any network architecture that results in this state space trajectory will give rise to the single-cell responses depicted in Figure 2. Figure 5 depicts four network topologies that result in this trajectory. The topologies can be

categorized as either using purely two-dimensional representations (Fig. 5*a*,*c*) or not (Fig. 5*b*,*d*), with behavior that is either entirely locally generated (Fig. 5*a*,*b*) or not (Fig. 5*c*,*d*). Note that all networks eventually rely on a two-dimensional representation to explain the results. We believe that this taxonomy of networks can both guide and be adjudicated by experimental results, as discussed. First, however, we turn to a brief characterization of each topology.

Two-dimensional integrator

Neural integration is a robust and common phenomenon across brain areas and has been widely associated with working memory (Douglas et al., 1995; Seung, 1996, 2000; Aksay, 2000, 2001). Because we are using a two-dimensional representation, it may make sense to account for this phenomena using a local, two-dimensional integrator.

In terms of the desired trajectory through the two-dimensional space, one dimension of the signal must represent the frequency, $F(t)$, and the other must represent a monotonically increasing function of time, $T(t)$, that is dependent on the initial frequency. $F(t)$ can be linked directly to the integration of f1, whereas $T(t)$ can be considered an integration of the integrated f1 signal. In other words, to reproduce the observed response curves, we must integrate both the input frequency (to remember it) and the results of that integration (to generate the ramping time signal).

These "double-integration" dynamics can be defined in standard control theoretic terms, using the dynamics equation, $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \alpha & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ 0 \end{bmatrix}, \qquad (6)$$

where u represents the stimulus presentation and α is a constant that scales the effect of the frequency component on the time component (i.e., the second integration). As described in the Appendix (see Neural dynamics), we can use Equation 13 to convert this two-dimensional integration network into a neurally implementable network:

$$\mathbf{A}' = \begin{bmatrix} 1 & 0 \\ \tau\alpha & 1 \end{bmatrix}, \mathbf{B}' = \begin{bmatrix} \tau & 0 \\ 0 & 0 \end{bmatrix}. \qquad (7)$$

As described in Results, this is a natural extension of previous characterizations of a one-dimensional integrator. The difference lies in the fact that the representation is two-dimensional and the dynamics are slightly more sophisticated, as reflected in the dynamics matrix $\mathbf{A}'$.

In our simulations of up to 3000 neurons, this circuit is unstable, because trajectories drift rapidly and then rest on one of a few attractor points (data not shown). These problems can be alleviated with longer time constants (which may be biophysically unrealistic), by adding neurons (Eliasmith and Anderson, 2003) (we were unable to pursue this solution further because of computational limitations), by including more sophisticated control circuits (e.g., a differentiating compensator), or by introducing varieties of hysteresis (Koulakov et al., 2002; Goldman et al., 2003).

Coupled one-dimensional integrators

Alternatively, it is possible to implement a similar solution in which the integration dynamics are handled in separate populations. Figure 5*b* shows the topology of such a network. Here, two one-dimensional integrators implement the same dynamic system (i.e., double integration) and project their results into a two-dimensional population. Nevertheless, Equation 6 describes the dynamics of this network as well, because only the representation and not the dynamics have changed. More accurately, Equation 6 could be written as two separate coupled differential equations (although this is mathematically equivalent). The reason two different architectures are described by the same equations is because anatomical constraints are important for determining the precise relationship between a dynamical description and its implementation in a set of neurons, although such constraints are not capture by the equations. So, the difference between panels *a* and *b* in Fig. 5 is the result of a difference in our assumptions regarding anatomical organization in these two cases (in the first that there are broad reciprocal projections from all two-dimensional neurons, in the second that there are feedforward projections from the one-dimensional integrators to the two-dimensional population).

The results from this network are shown in Figure 2*A–F*. To achieve these results, each integrator used 1000 one-dimensional neurons, and the two-dimensional population consisted of 500 neurons. The network is highly stable and thus able to reproduce the results from the experiment. Notably, both this solution and the previous one assume that the signals driving the movement through the two-dimensional state space are internally generated, which is consistent with the idea that the network itself is keeping track of the time elapsed between stimulus presentations.

Projected external signals

Unlike the characterization of the previous two architectures, the dynamics could be driven by external signal sources. For instance, signals from the ventral tegmental area project to the PFC and could account for the time-varying signal observed in working memory. However, the existence of ramping signals seem to be sensitive to the uncertainty regarding a reward, whereas the monkey in these experiments receive a reward on almost all trials. Indeed, it is possible that both the $F(t)$ and $T(t)$ signals are external to the population in the PFC. In this scenario "working memory" is a misnomer for the function of this area, because there is no active maintenance of a local signal in the region. Rather, the PFC would simply be a merging of two independent signals into a single two-dimensional representation. The results from this network are identical to those presented in Figure 2, given appropriate input signals.

Including such external signals can be accomplished with a simple modification of Equation 6:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ e \end{bmatrix}. \qquad (8)$$

With $\alpha_{1,2} < 0$, neither dimension will be integrated as it was in the previous networks and will thus both reflect the external signals directly.

One-dimensional integrator with one external input

If, however, $\alpha_1 = 0$ or $\alpha_2 = 0$, then the corresponding dimension will be integrated. Thus if only the frequency or time signals were projected into the network from an external source, appropriately varying Equation 8 will reflect this fact. For instance, supposing that $\alpha_1 = 0$, then u would reflect the standard input to working memory that would be integrated in this area, and e would reflect an externally generated ramping time signal. The results of simulating this network are identical to those in Figure 2 given appropriately chosen external signals for the chosen dynamics.

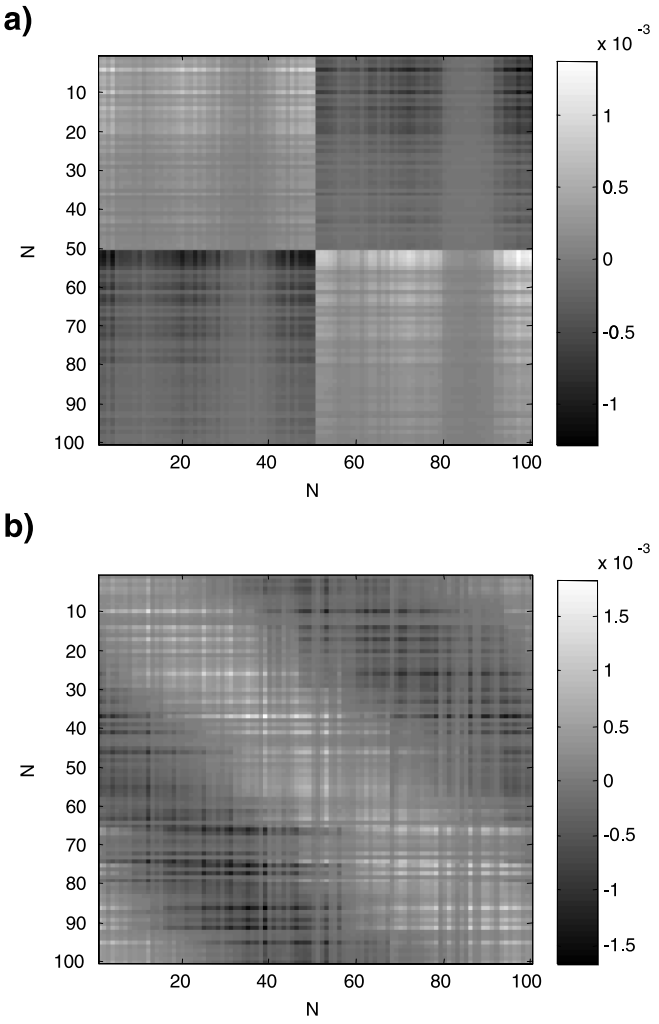

Figure 6. Recurrent connection weights for the one-dimensional integrator (***a***) and two-dimensional integrator (***b***). Both axes are *N*, the neuron number assigned after sorting the neurons by their encoders and decoders. The depth of gray indicates connection strength. These matrices demonstrate the systematic relationship between center-surround connectivity and attractor networks of any dimension.

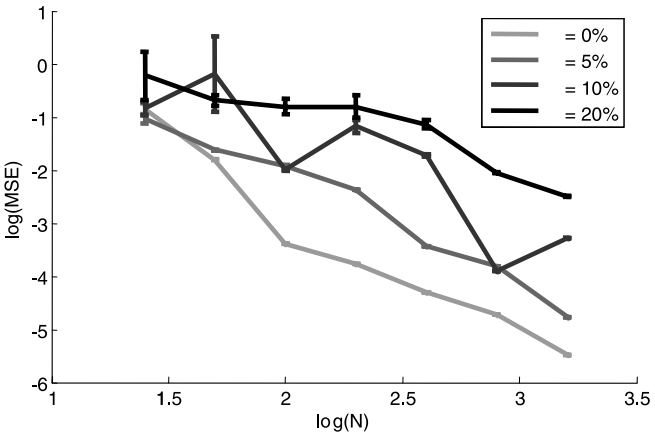

Figure 7. Network robustness to connection weight noise. This is a log–log plot of the MSE as a function of the number of neurons (*N*) and the amount of Gaussian noise added to the weights in the connection matrix. The noise added is calculated by taking the indicated percentage as the SD of Gaussian, independent mean zero noise that is then scaled by the original size of the weight.

General dynamics results

Despite this variety of possible architectures, there are some general insights that can be gained from examining them together (we discuss ways of empirically distinguishing these topologies in the Discussion). First, it is useful to recall at this point that the connection matrices for all dynamics in these various architectures are of the same form (i.e., $\omega_{ij} = \alpha_1 \langle \tilde{\phi}_i \mathbf{A}' \phi_j^{\mathbf{x}} \rangle$ for recurrent connections and $\omega_{ij} = \alpha_1 \langle \tilde{\phi}_i \mathbf{B}' \phi_k^{\mathbf{u}} \rangle$ for input connections). Obviously, the specific values of these variables will change the precise structure of the matrices. Nevertheless, there is some degree of "typical" structure for recurrently connected integrators, regardless of their dimension (Fig. 6*a*,*b*). Specifically, both one- and two-dimensional integrator populations display a noisy center-surround organization, although this pattern is more evident in the two-dimensional case. For the one-dimensional case, there are only two possible encoders (± 1), so the diagonal center is the size of half of the population. However, this center-surround structure has been observed in higher dimensions (e.g., 25 dimensions) as well (Conklin and Eliasmith, 2005). And this pattern is strikingly similar to the hand-constructed center-surround weight matrices used in past integrator models (Zhang, 1999). As a result, if connectivity patterns of neural populations resemble this general pattern, they may be involved in a form of neural integration for any dimensionality of representation.

One concern that arises with constructing weight matrices using these methods is the "fine-tuning problem" (i.e., the problem of making the matrix robust to noise). One difficulty with this characterization of the problem is that whether or not there is actually a problem depends on the precise kind of disturbance introduced. The weights in these networks are clearly robust to some noise, as demonstrated by Figure 7. This figure shows that a desired mean squared error (MSE) can be reached by increasing the size of the integration population. Specifically, it can be seen that the effects of the noise go down as $\sim 1/N$. These results indicate that the integrator will be robust even for small population sizes, because integrator stability is directly related to the MSE (Eliasmith and Anderson, 2003). Similar robustness results for a high-dimensional integrator have been reported by Conklin and Eliasmith (2005), who showed that integration behavior was only mildly affected by the addition of up to $\sigma = 50\%$ noise.

However, this robustness is to zero mean, independent Gaussian noise. We have no reason to believe that our network will be especially robust to non-zero mean noise in the connection weights. What is of importance, then, is not the presence of noise, but rather some population-wide bias (e.g., shifting all weights in one direction). We are unaware of any experimental results indicating that such biases should be expected here, or in any other integrator networks. We suggest that fine-tuning is thus a misnomer for the problem (because we can add random noise the weights to a large degree and the network still functions). Perhaps it should be called the "biased weight problem" instead.

Discussion

There are three major insights to be drawn from these results. First, a two-dimensional neural representation can underwrite a natural, efficient, and robust network that explains the wide variety of responses exhibited by working memory neurons. In this regard, it is important to note that this approach is also relevant for systems with nonmonotonic (e.g., peaked) tuning curves, which are observed in many working memory tasks (Nieder et al., 2002). This is because if the dimensions of the representation are polar coordinates (rather than Cartesian), peaked tuning naturally results (because a sweep around the θ dimension produces a

peak centered on the preferred-direction vector), and the dynamics can be described analogously. Notably, higher-dimensional models can also result in this kind of peaked tuning (Eliasmith and Anderson, 2003; Conklin and Eliasmith, 2005). These representations may be relevant for understanding working memory in visual areas, which are sensitive to a large number of dimensions.

Second, this explanation relies on the observed heterogeneity of neural responses. Elsewhere, we have argued that neural heterogeneity is a natural result of a trade off between representational efficiency and simple organizational mechanisms in neural systems (Eliasmith and Anderson, 2003). So, this work supports the idea that, rather than being a sign of the intractability of neural systems, heterogeneity is both consistent with efficient neural computation and essential to incorporate into models attempting to explain such computation.

Third, these results provide evidence that the methods used here allow for effective, plausible, and quantitative characterizations of neural representation and neural dynamics resulting from a wide variety of possible network topologies.

However, these conclusions do not directly address two central empirical questions: (1) How might we tell whether working memory uses a two-dimensional rather than one-dimensional (or higher-dimensional) representation? (2) How can we adjudicate between the possible network topologies that give rise to the state space trajectory?

Regarding the first question, we note that one benefit of a two-dimensional representation is that it supports the linear extraction of nonlinear functions of the represented dimensions (Eliasmith and Anderson, 2003). If these neurons are used to encode not only stimulus parameters but also act as a kind of preparatory signal [as suggested by Brody et al. (2003)], extracting such a function is essential. So, the model leads to the prediction that some neural populations that receive projections from this area will compute nonlinear functions of the represented variables (which should be evident in their tuning). This highlights the close link between nonlinear computation and higher-dimensional representation (under the assumption that neural populations in the cortex perform linear transformations, i.e., by summing weighted synaptic input).

However, this does not address the question of whether the population may be representing more than two dimensions. This is a valid concern because our simulation does not display all of the subtleties of the data. For example, the minor ramping of early neurons after the first second (Fig. 2*a*,*b*) are absent in our simulations (Fig. 2*A*,*B*). Additionally, we do not find the "down-then-up" responses evident in some neurons (data not shown). Despite the fact that these responses account for a small portion of the categorized curves, higher-dimensional extensions to the current model might account for these additional phenomena. A principal, or independent, component analysis of the experimental data could provide a good indication of how many dimensions are required. It is important to note, however, that such features of the data might also arise from different dynamics: including the state space trajectory (we have assumed a very simple one).

Considering the number of dimensions needed to model the data highlights predictions that distinguish models of different dimensions. From a purely mathematical point of view, any D-dimensional representation can be reproduced by D one-dimensional representations (because both representations span the D-dimensional space). However, when there are resource constraints (e.g., maximum firing rates), these two representations can be distinguished because saturation effects will be different. Consider two two-dimensional populations: the first has all preferred-direction vectors aligned with the *x*- or *y*-axes (because it is equivalent to two one-dimensional populations, we call it the one-dimensional model), and the second has preferred-direction vectors evenly distributed over the two-dimensional space (the two-dimensional model). Distributions between these two extremes will show related effects to greater and lesser degrees. In the first case, the saturation of representations along the *x* dimension will be unaffected by any variations in the representations along the *y* dimension. In the second case, for the majority of preferred-direction vectors, any increase in *x*-related firing results in a comparable reduction in the range of *y* dimension values that can be represented before saturation. These effects will likely only be observed for delay periods longer than the expected delay because of rapid renormalization of the stimuli. If the renormalization is rapid enough, it may only be the first trial after that expectation is violated that shows the effects. These effects should be both neurally and behaviorally observable.

Consider high-frequency trials and neurons with positive monotic, downward ramping responses. In the two-dimensional model, such neurons are nearly saturated by the frequency input but are driven toward zero by the ramping time signal. This sets up a conflict between the representation of frequency and time. So, after the saturation of the time signal (i.e., in which the time signal is very negative for the neurons), these neurons will nevertheless have an above background firing rate because they are also contributing to the representation of frequency. As a result, they will downward slope but level off, even with an increasing time signal. In contrast, a one-dimensional model will have all downward ramping neurons driven to very low or zero firing rates. This is because they are only representing the temporal aspect of the signal, so there is no source of current (e.g., from representing frequency) to counteract the strong downward signal.

Behaviorally there should be differences as well. In the trials that violate the expected delay period, there should be a progressive worsening of accuracy of response as the time course goes past the expected delay period, because the *x* and *y* dimensions interact, and *y* is ramping. However, this accuracy effect should level off once both dimensions are saturated. In contrast, in a one-dimensional model, there should be no effect on accuracy of changes in time delay, because the dimensions represent (and therefore saturate) independently.

Turning to the second question (adjudication of network dynamics), distinguishing these network topologies could be accomplished through carefully constructed microstimulation experiments. This methodology has been applied to work in decision making, motion processing, eye control, and tactile working memory (Cohen and Newsome, 2004). Although there are some concerns regarding the application and effects of stimulation, the ability of microstimulation to elicit equivalent responses to tactile stimuli in S1 (Romo et al., 1998) suggests the somatosensory cortex may be a good target for microstimulation. As well, microstimulation has been applied successfully to characterizing neural integration, a closely related form of dynamics (Kustov and Robinson, 1995).

Microstimulation could be highly informative regarding the network topology both for distinguishing the origin of the signals, and for distinguishing one- and two-dimensional integration. In the first case, if the memory signal is not stored in this anatomical area but rather is projected to the network, brief stimulation should only temporarily affect the representation during the delay period. If, however, stimulation resulted in disruption of the dynamics, and hence performance on the task, then the

signals are at least partially locally generated. The differential effects of microstimulation on one- or two-dimensional integrators is likely more subtle. If both signals are disrupted after stimulation and there is a systematic relationship between the two disruptions (e.g., the time signal is still the integral of the frequency signal), then Figure 5*b* is most likely. In contrast, if both signals are arbitrarily disrupted by stimulation, Figure 5*a* is more likely because the representation and dynamics in both dimensions will be changed concurrently. If only one or the other signal is locally generated, then only that signal would be disrupted by microstimulation. So, it would be difficult to distinguish panel *b* from panel *d* in Figure 5, because both are consistent with that outcome. Thus, a series of results in which either the time signal is disrupted or both signals are disrupted would be required to make Figure 5*b* more likely. If only one of the signals was ever disrupted, then Figure 5*d* is more likely. We should note that this reasoning is dependent on the assumption of very small backpropagation effects during microstimulation. It is commonly assumed that microstimulation is highly local, affecting an area of only ~200 μm or a few hundred cells (Tu and Keating, 2000; Cohen and Newsome, 2004). However, some studies suggest the possibility that the effects could traverse hemispheres (Seidemann et al., 2002).

In conclusion, our simulations demonstrate that a diverse population of two-dimensional neurons can naturally reproduce the data of Romo et al. (1999). We have emphasized how the methods exploited here can suggest ways of systematically exploring the possible topologies to generate these results. Doing so makes it clear that limiting models to one-dimensional representations unnecessarily limits the hypotheses being considered and can overcomplicate models of the observed responses. Given general methods for building models with higher-dimensional populations, and given the availability of data that are highly suggestive of trajectories through higher-dimensional spaces, there is good reason to adopt this kind of alternative explanation of neural behavior.

Appendix

Neural representation

We take representation in neural populations to be characterized in terms of a nonlinear encoding process and a linear decoding process (Eliasmith and Anderson, 2003). Encoding involves converting a quantity, $\mathbf{x}(t)$, into a spike train:

$$\sum_n \delta(t - t_{in}) = G_i[J_i(\mathbf{x}(t))], \tag{9}$$

where $G_i[\cdot]$ is the nonlinear function describing the spiking response (see Figs. 3 and 4*b* for typical LIF responses), J_i is the current in the soma of the cell, *i* indexes the neuron, and *n* indexes the spikes produced by the neuron. Note that the driving current is described in detail by Equation 1 and the nonlinearity is described by Equation 2. Equation 9 captures the nonlinear encoding process from a high-dimensional variable, $\mathbf{x}$, to a one-dimensional soma current, J_i, to a train of spikes, $\delta(t - t_{in})$.

To understand how a neural system might use the information encoded into a spike train in this manner, we must characterize a neurally plausible decoding as well. To do so, we need to understand how this information can be converted from spike trains back into a relevant quantity. Note that we are not suggesting that this decoding process takes place explicitly in neurons. Rather, it is a theoretically useful means of characterizing part of the information processing characteristics of neurons. In particular, we characterize decoding in terms of PSCs and connection weights. Somewhat surprisingly, a plausible means of characterizing this decoding is as a linear transformation of the spike train. Specifically, we can estimate the original stimulus vector $\mathbf{x}(t)$ by decoding an estimate, $\hat{\mathbf{x}}(t)$, using a linear combination of filters, $h_i(t)$, weighted by decoding weights, ϕ_i, as follows:

$$\hat{\mathbf{x}}(t) = \sum_{in} \delta(t - t_{in}) * h_i(t)\phi_i = \sum_{in} h_i(t - t_{in})\phi_i, \tag{10}$$

where * indicates convolution (see Fig. 1). These $h_i(t)$ are thus linear decoding filters that, for reasons of biological plausibility, we take to be the PSCs in the subsequent neuron.

Revisiting Figure 1, we can understand the depicted processes in terms of these equations. The encoding in Figure 1*b* produces a raster of spike trains, $\delta(t - t_{in})$, where t_{in} indicates the *n*th spike for neuron *i*. Neurons are separated at $i = 50$ into on and off neurons ($\phi_i = +1$ and -1, respectively) and sorted by firing onset (the value of $\mathbf{x}$ for which $G_i[J_i(\mathbf{x})] = 0$). In Figure 1*c*, spike trains are plotted with their PSC-filtered counterpart [i.e., $h_i(t - t_{in}) = \delta(t - t_{in}) * h_i(t)$]. Finally, the weighted sum of all of the filtered trains, $\Sigma_{in} h_i(t - t_{in})\phi_i$, yields the overall decoded estimate (Fig. 1*a*, black line).

To find the ϕ_i weights to determine this estimate, we minimize the mean-squared error (see also Salinas and Abbott, 1994; Seung et al., 2000):

$$E = \frac{1}{2}\langle[\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^2\rangle_{\mathbf{x},t}$$

$$= \frac{1}{2}\langle[\mathbf{x}(t) - \sum_{in}(h_i(t - t_{in}) + \eta_i)\phi_i]^2\rangle_{\mathbf{x},t,\eta} \tag{11}$$

where $\langle\cdot\rangle_x$ denotes integration over the range of $\mathbf{x}$ and η_i models the expected noise. By optimizing with Gaussian random noise, we ensure that fine tuning is not a concern, because the decoding weights will be robust to fluctuations.

This method provides a means of defining *n*-dimensional representations in a biologically plausible population of neurons. Here, we have taken the population, ϕ_i, and temporal, $h_i(t)$, decoders to be independent, although this is not necessary, as can be seen from the fact that Equation 11 can also be minimized over time.

Neural dynamics

For generality, we can write the relevant dynamics of a population in a control theoretic form (i.e., using the dynamics state equation that comprises the foundation of modern control theory):

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{12}$$

where $\mathbf{A}$ is the dynamics matrix, $\mathbf{B}$ is the input matrix, $\mathbf{u}(t)$ is the input or control vector, and $\mathbf{x}(t)$ is the state vector (see Fig. 8*a* for a graphical depiction of this equation). In general, these matrices and vectors can describe a wide variety of linear, time-invariant physical systems [Eliasmith and Anderson (2003) show how these same methods apply to time-varying and nonlinear systems as well]. Here, we use Equation 12 to capture the hypothesized high-level dynamics of a population of neurons.

Initially, this high-level characterization is divorced from neural-level, implementational considerations. However, it is possible to modify these matrices to render the system neurally plausible. First, we must account for intrinsic neural dynamics by converting this characterization into a neurally relevant one (Fig.

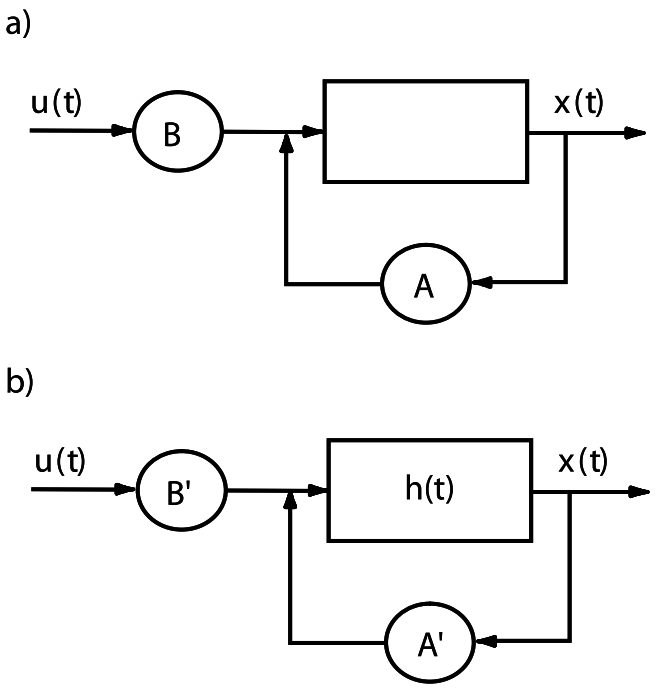

Figure 8. Control theoretic block diagram for time invariant linear systems. ***a***, Diagram of the standard state equation, $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$. ***b***, Diagram of the neural state equation, $\mathbf{x}(t) = h(t) * [\mathbf{A}'\mathbf{x}(t) + \mathbf{B}'\mathbf{u}(t)]$. Note that the dot is dropped in the neural equation because the dynamics of the filter, $h(t)$, accounts for integration. We can convert ***a*** into ***b*** using Equation 13.

8*b*). To do so, we assume a model of PSCs given by $h(t) = \tau^{-1}e^{-t/\tau}$ and can derive the following relationship between panels *a* and *b* in Figure 8 [see Eliasmith and Anderson (2003) for a discussion that justifies assuming this (or, more generally, any typically "peaked") PSC model]:

$$\mathbf{A}' = \tau\mathbf{A} + \mathbf{I}.$$
$$\mathbf{B}' = \tau\mathbf{B} \quad (13)$$

So, our description of the high-level neurally plausible dynamics becomes the following:

$$\mathbf{x}(t) = h(t)^*[\mathbf{A}'\mathbf{x}(t) + \mathbf{B}'\mathbf{u}(t)]. \quad (14)$$

Notably, this transformation is general and assumes nothing about the form of **A** or **B**. So, given any behavioral system defined in the form of Equation 12, it is possible to construct the neural counterpart by solving for **A′** and **B′**. A variety of applications of this method to linear, nonlinear, and time-varying neural systems is described by Eliasmith (2005).

Next, we must incorporate this high-level description of the dynamics with our previous characterization of the neural representation. To do so, we combine the dynamics of Equation 14, the encoding of Equation 9, and the population decoding of **x** and **u** from Equation 10. That is, we take $\hat{\mathbf{x}} = \Sigma_{\text{jn}} h_j (t - t_{\text{jn}})\phi_j^{\mathbf{x}}$ and $\hat{\mathbf{u}} = \Sigma_{\text{kn}} h_{\text{k}} (t - t_{\text{kn}})\phi_{\text{k}}^{\mathbf{u}}$, which gives the following:

$$\sum_n \delta(t - t_{\text{in}}) = G_{\text{i}}[\alpha_{\text{i}}\langle\tilde{\phi}_{\text{i}}\mathbf{x}(t)\rangle + J_{\text{i}}^{bias}]$$
$$= G_{\text{i}}[\alpha_{\text{i}}\langle\tilde{\phi}_{\text{i}}[\mathbf{A}'\hat{\mathbf{x}}(t) + \mathbf{B}'\hat{\mathbf{u}}(t)]\rangle + J_{\text{i}}^{bias}]$$
$$= G_{\text{i}}[\alpha_{\text{i}}\langle\tilde{\phi}_{\text{i}}[\mathbf{A}'\Sigma_{\text{jn}}h_j(t - t_{\text{jn}})\phi_j^{\mathbf{x}} + \mathbf{B}'\Sigma_{\text{kn}}h_{\text{k}}(t - t_{\text{kn}})\phi_{\mathbf{ku}}]\rangle + J_{\text{i}}^{bias}]. \quad (15)$$

It is important to keep in mind that the temporal filtering is only done once (here included in the estimate of the signals), despite the fact that it is include in both Equations 14 and 10. That is, $h(t)$ in these equations both defines the dynamics and defines the decoding of the representations. To put it in a more familiar form, this equation can be written as follows:

$$G_{\text{i}}[\alpha_{\text{i}}\langle\tilde{\phi}_{\text{i}}[\mathbf{A}'\Sigma_{\text{jn}}h_j(t - t_{\text{jn}})\phi_j^{\mathbf{x}} + \mathbf{B}'\Sigma_{\text{kn}}h_{\text{k}}(t - t_{\text{kn}})\phi_{\text{k}}^{\mathbf{u}}]\rangle + J_{\text{i}}^{bias}]$$
$$= G_{\text{i}}[\Sigma_{\text{jn}}\omega_{\text{ij}}h_j(t - t_{\text{jn}}) + \Sigma_{\text{kn}}\omega_{\text{ik}}h_{\text{k}}(t - t_{\text{kn}}) + J_{\text{i}}^{bias}], \quad (16)$$

where $\omega_{\text{ij}} = \alpha_1\langle\tilde{\phi}_{\text{i}}\mathbf{A}'\phi_j^{\mathbf{x}}\rangle$ and $\omega_{\text{ik}} = \alpha_1\langle\tilde{\phi}_{\text{i}}\mathbf{B}'\phi_{\text{k}}^{\mathbf{u}}\rangle$ are the recurrent and input connection weights, respectively. These weights will now implement the dynamics defined by the control theoretic structure from Equation 14 in a neurally plausible network.

References

Aksay E, Baker R, Seung HS, Tank DW (2000) Anatomy and discharge properties of pre-motor neurons in the goldfish medulla that have eye-position signals during fixations. J Neurophysiol 84:1035–1049.

Aksay E, Gamkrelidze G, Seung HS, Baker R, Tank DW (2001) In vivo intracellular recording and perturbation of persistent activity in a neural integrator. Nat Neurosci 4:184–193.

Brody CD, Romo R, Kepecs A (2003) Basic mechanisms for graded persistent activity: discrete attractors, continuous attractors, and dynamic representations. Cogn Neurosci 13:204–211.

Camperi M, Wang X-J (1998) A model of visuospatial short-term memory in prefrontal cortex: cellular bistability and recurrent network. J Comput Neurosci 5:383–405.

Cohen MR, Newsome WT (2004) What electrical microstimulation has revealed about the neural basis of cognition. Curr Opin Neurobiol 14:1–9.

Conklin J, Eliasmith C (2005) An attractor network model of path integration in the rat. J Comp Neurosci 18:183–203.

Douglas R, Koch C, Mahowald M, Martin K, Suarez H (1995) Recurrent excitation in neocortical circuits. Science 269:981–985.

Eliasmith C (2005) A unified approach to building and controlling spiking attractor networks. Neural Comp 17:1276–1314.

Eliasmith C, Anderson C (2003) Neural engineering: computation, representation, and dynamics in neurobiological systems. Cambridge, MA: MIT.

Fiorillo CD, Tobler PN, Schultz W (2003) Discrete coding of reward probability and uncertainty by dopamine neurons. Science 299:1898–1902.

Funahashi S, Bruce CJ, Goldman-Rakic PS (1989) Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. J Neurophysiol 61:331–349.

Fuster JM (1973) Unit activity in the prefrontal cortex during delayed response performance: neuronal correlates of short-term memory. J Neurophysiol 36:61–78.

Georgopoulos AP, Kalasaka JF, Crutcher MD, Caminiti R, Massey JT (1984) The representation of movement direction in the motor cortex: single cell and population studies. In: Dynamic aspects of neocortical function (Edeleman GM, Gail WE, Cowan WM, eds), pp 501–524. New York: Neurosciences Research Foundation.

Gnadt JW, Andersen RA (1988) Memory related motor planning activity in posterior parietal cortex of macaque. Exp Brain Res 70:216–220.

Goldman MS, Levine JH, Major G, Tank DW, Seung HS (2003) Robust persistent neural activity in a model integrator with multiple hysteretic dendrites per neuron. Cereb Cortex 13:1185–1195.

Koch C (1999) Biophysics of computation: information processing in single neurons. New York: Oxford UP.

Koulakov AA, Raghavachari S, Kepecs A, Lisman JE (2002) Model for a robust neural integrator. Nat Neurosci 5:775–782.

Kustov AA, Robinson DL (1995) Modified saccades evoked by stimulation of the macaque superior colliculus account for properties of the resettable integrator. J Neurophysiol 73:1724–1728.

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

Nieder A, Freedman DJ, Miller EK (2002) Representation of the quantity of visual items in the primate prefrontal cortex. Science 297:1708–1711.

Reutimann J, Yakovlev V, Fusi S, Senn W (2004) Climbing neuronal activity as an event-based cortical representation of time. J Neurosci 24:3295–3303.

Romo R, Hernandez A, Zainos A, Salinas E (1998) Somatosensory discrimination based on cortical microstimulation. Nature 392:387–390.

Romo R, Brody C, Hernandez A, Lemus L (1999) Neuronal correlates of parametric working memory in the prefrontal cortex. Nature 399:470–473.

Salinas E, Abbott L (1994) Vector reconstruction from firing rates. J Comp Neurosci 1:89–107.

Seidemann E, Arieli A, Grinvald A, Slovin H (2002) Dynamics of depolarization and hyperpolarization in the frontal cortex and saccade goal. Science 295:862–865.

Seung HS (1996) How the brain keeps the eyes still. Proc Natl Acad Sci USA 93:13339–13344.

Seung HS, Lee DD, Reis BY, Tank DW (2000) Stability of the memory of eye position in a recurrent network of conductance-based model neurons. Neuron 26:259–271.

Taube JS, Bassett JP (2003) Persistent neural activity in head direction cells. Cereb Cortex 13:1162–1172.

Tu TA, Keathing EG (2000) Electrical stimulation of the frontal eye field in a monkey produces combined eye and head movements. J Neurophysiol 84:1103–1106.

Wang XJ (1999) Synaptic basis of cortical persistent activity: the importance of NMDA receptors to working memory. J Neurosci 19:9587–9603.

Zhang K (1999) Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. J Neurosci 16:2112–2126.

Zipser D, Kehoe B, Littlewort G, Fuster J (1993) A spiking network model of short-term active memory. J Neurosci 13:3406–3420.