LETTER **Communicated by Xiao-Jing Wang**

Design of Continuous Attractor Networks with Monotonic Tuning Using a Symmetry Principle

Christian K. Machens
machens@zi.biologie.uni-muenchen.de
Carlos D. Brody
brody@princeton.edu
Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, U.S.A.

Neurons that sustain elevated firing in the absence of stimuli have been found in many neural systems. In graded persistent activity, neurons can sustain firing at many levels, suggesting a widely found type of network dynamics in which networks can relax to any one of a continuum of stationary states. The reproduction of these findings in model networks of nonlinear neurons has turned out to be nontrivial. A particularly insightful model has been the "bump attractor," in which a continuous attractor emerges through an underlying symmetry in the network connectivity matrix. This model, however, cannot account for data in which the persistent firing of neurons is a monotonic—rather than a bell-shaped—function of a stored variable. Here, we show that the symmetry used in the bump attractor network can be employed to create a whole family of continuous attractor networks, including those with monotonic tuning. Our design is based on tuning the external inputs to networks that have a connectivity matrix with Toeplitz symmetry. In particular, we provide a complete analytical solution of a line attractor network with monotonic tuning and show that for many other networks, the numerical tuning of synaptic weights reduces to the computation of a single parameter.

1 Introduction

Animals are constantly confronted with the task of storing and manipulating continuous variables, such as the spatial position of objects or body parts. The short-term storage of such information is thought to be mediated by persistently active neurons in various areas of the brain (Major & Tank, 2004; Wang, 2001). For instance, neurons in the oculomotor system fire in proportion to an animal's eye position (Aksay, Gemkrelidze, Seung, Baker, & Tank, 2001), neurons in the head-direction system fire with respect to an animal's directional heading (Taube & Bassett, 2003), and neurons in the hippocampus of rats fire with respect to the location of an animal (O'Keefe, 1979; Quirk, Muller, & Kubie, 1990). All of these firing rates persist in the absence of proprioreceptive or other sensory information, suggesting that they store information about the respective analog variables. Further examples come from many cortical areas in which neural activities can represent information about remembered sensory variables, such as the frequency of a vibratory stimulus (Romo, Brody, Hernandez, & Lemus, 1999) or a spatial location (Smyrnis, Taira, Ashe, & Georgopoulos, 1992; Nakamura, 1999).

In all of the cited examples, the firing rates of the neurons are continuous, smoothly varying functions of the stored analog variable. These functions (or tuning curves) seem to come in two distinct classes. In one class, the firing rates are bell-shaped or bump-shaped functions of the stored variable. Such tuning curves have been prominent for periodic angular variables such as head direction (Taube & Bassett, 2003), but have also been found for nonperiodic, nonangular variables such as the spatial position of an animal (O'Keefe, 1979; Quirk et al., 1990). In the other class, the firing rates are monotonic functions of the stored variable. Such tuning curves have been observed in the oculomotor system (Seung, Lee, Reis, & Tank, 2000) and the prefrontal cortex (Romo et al., 1999).

The ability to maintain firing rates at different constant levels, often referred to as *graded* persistent activity, suggests that the respective neural systems are able to maintain a continuum of stationary states, each corresponding to a particular pattern of firing rates and a particular memory value. Such a continuum of stationary states is usually called a continuous attractor or line attractor of the respective neural dynamical system.

Neural network models of line attractors, built to account for the experimental data described above, are often based on control of precise feedback in the recurrent connections between neurons. (But see Loewenstein & Sompolinsky, 2003, and Fransen, Tahvildari, Egorov, Hasselmo, & Alonso, 2006, for single-cell models of persistent activity.) Designing such attractor networks using neurons with linear input-output functions is fairly straightforward. However, when realistic nonlinear neurons are considered, finding network connectivities that yield continuous attractors becomes a nontrivial problem. A particularly elegant solution exists for the case of bell-shaped tuning curves. In the so-called bump attractor, the attractor property is derived from an underlying symmetry in the network connectivity, a translational invariance in the ordering of synaptic weights (Amari, 1977; Ben-Yishai, Bar-Or, & Sompolinsky, 1995; Skaggs, Knierim, Kudrimoti, & McNaughton, 1995; Zhang, 1996; Samsonovich & McNaughton, 1997; Tsodyks, 1999; Compte, Brunel, Goldman-Rakic, & Wang, 2000). In contrast, an exact solution for the construction of realistic network models with monotonic tuning curves has been evasive. Instead, researchers building such network models have relied on computationally intensive numerical tuning to approximate a line attractor (Seung et al., 2000; Koulakov, Raghavachari, Kepecs, & Lisman, 2002; Miller, Brody, Romo, & Wang, 2003; Eliasmith, 2005).

In this letter, we show that the construction of monotonic line attractors can be based on the same symmetry principles used in the bump attractor. In particular, we provide an exact and analytical solution of a monotonic line attractor network built out of nonlinear neurons and show that this network and the bump attractor network are instantiations of a whole family of line attractor networks based on weight matrices with a Toeplitz symmetry. By connecting the emergence of a continuous attractor with an underlying symmetry in the network, we gain insights that are less easily obtained with numerical approaches. We finally show how the monotonic line attractors based on Toeplitz symmetry are related to the monotonic line attractors that have been designed earlier (Seung, 1996; Koulakov et al., 2002; Miller et al., 2003; Eliasmith, 2005).

2 Network Dynamics in The Mean Field Limit

In the experimental data described above, analog variables are represented by the firing rates of neurons, and these firing rates stay approximately constant over several seconds. We will therefore use firing-rate-based models to approach the network design problem (Wilson & Cowan, 1973; Amari, 1977; Dayan & Abbott, 2001). We do note that in some experimental systems, persistent neural activity can be seen to be slowly but significantly time varying (Brody, Hernandez, Zainos, & Romo, 2003; Rainer & Miller, 2002), and some network models have begun to incorporate this aspect (Miller et al., 2003; Durstewitz, 2003; Singh & Eliasmith, 2006). However, for simplicity we will focus here on the case of networks with non-time-varying persistent activity. In other words, we are concerned with stationary states—fixed points of the network dynamics.

We label each neuron by a position label x and define $s_{\mathrm{tot}}(x, t)$ to be the total synaptic input to neuron x at time t. We further define $r(x, t)$ to be the firing rate of neuron x at time t. Its dependence on synaptic input is described by a function $h(\cdot)$, determined by the biophysics of single neurons, such that $r(x, t) \equiv h\big(s_{\mathrm{tot}}(x, t)\big)$. The synaptic output of neuron x, defined as the fraction of the maximal conductance of its postsynaptic targets, depends on the neuron's firing rate r and is given by a function $g(\cdot)$ that is determined by synaptic properties: $s(x, t) \equiv g\big(r(x, t)\big) \in [0, 1]$. (see appendix A for details). We label the composition of these two functions, which goes from synaptic input to synaptic output, as $f(\cdot) \equiv g\big(h(\cdot)\big)$.

Let $w(x, x')$ be the total strength of connections from neuron x' to neuron x. We assume that different inputs to a neuron combine additively. For simplicity, we assume that the effect of different synapse types can be incorporated in the weight matrix $w(x, x')$. In particular, we assume that excitatory inputs correspond to positive weights and inhibitory inputs to negative weights. Finally, let $E(x)$ denote external synaptic inputs to neuron x, coming from neurons or areas of the brain that are outside this network.

We will assume that these external inputs are constant on the timescales that interest us.

Under these assumptions, the standard mean field equation for continuous neural networks is given by (Wilson & Cowan, 1973; Dayan & Abbott, 2001)

$$\frac{\partial}{\partial t}s(x,t) = -s(x,t) + f\left(\int_{-1}^{1} dx' w(x,x')s(x',t) + E(x)\right). \tag{2.1}$$

In the usual interpretation of equation 2.1, $s(x)$ is the firing rate of neuron x. In our interpretation, however, s will be the synaptic output of a neuron and is related to the firing rate r through $s = g(r)$. With this interpretation, the above formulation of the network dynamics is equivalent to the dynamics of a conductance-based network of asynchronously spiking neurons (Ermentrout, 1994; Seung et al., 2000; Shriki, Hansel, & Sompolinsky, 2003; Renart, Brunel, & Wang, 2003).

3 Line Attractors Reconsidered

3.1 Definition. Stationary network states correspond to the case $\partial_t s(x,t) = 0$ in equation 2.1 so that

$$s(x) = f\left(\int_{-1}^{1} dx' w(x,x')s(x') + E(x)\right). \tag{3.1}$$

Given a net input-output function $f(\cdot)$, defined by the biophysical properties of neurons, our design goal is to find weight matrices $w(x,x')$ and external inputs $E(x)$ that generate a continuum of solutions satisfying equation 3.1. If we denote these solutions with a parameter $m \in [m_0, m_1]$, then

$$s(x,m) = f\left(\int_{-1}^{1} dx' w(x,x')s(x',m) + E(x)\right). \tag{3.2}$$

The value of m corresponds to the memory stored in the network. As a function of m, the network states $s(x,m)$ satisfying equation 3.2 trace out a continuous line in the space of all network states. A one-dimensional continuous attractor is therefore also called a line attractor.

Since $f(\cdot)$ is a nonlinear function, the design problem—that of finding weight matrices $w(x,x')$ and external inputs $E(x)$ that fulfill equation 3.2—is nontrivial. In contrast, for linear input-output functions $f(\cdot)$, the class of possible solutions, not treated here, is readily obtained (Dayan & Abbott, 2001). Finding weight matrices and external inputs so that equation 3.2 holds does not guarantee that the resulting continuum of stationary states is dynamically stable. However, we will treat the stability of these solutions as a separate issue.

For a given neuron at position $x = x_0$, the function $s_0(m) = s(x_0, m)$ corresponds to the synaptic output of that neuron as a function of the memory m. Accordingly, the firing rate

$$r_0(m) = h\left(\int_{-1}^{1} dx' w(x_0, x')s(x', m) + E(x)\right) \tag{3.3}$$

corresponds to the tuning curve of the neuron (i.e., its firing rate as a function of m). Note that the tuning curves are defined via the function $r = h(s_{\text{tot}})$, the mapping of synaptic inputs into the firing rate, and not the input-output function $s = f(s_{\text{tot}})$ used in equations 3.1 and 3.2. The tuning curves provide the link to experimental data since they can be measured in electrophysiological experiments. An important constraint on line attractor design is therefore to find solutions to equations 3.2 and 3.3 that reproduce the distribution of tuning curves found in actual neural systems.

3.2 The Bump Attractor. The bump attractor is one of the first line attractor networks constructed with nonlinear elements (Amari, 1977; Ben-Yishai et al., 1995; Skaggs et al., 1995; Zhang, 1996; Samsonovich & McNaughton, 1997; Tsodyks, 1999; Compte et al., 2000; see also Figure 1). It has been thoroughly studied and yields some crucial insights into how a continuum of stationary states can emerge through network interactions. We briefly review the ideas that underlie the construction of the bump attractor here in order to pave the way for our own developments explored further below.

In its simplest form, the bump attractor consists of an infinite one-dimensional layer of neurons, $x \in \mathbb{R}$, that receive a spatially homogeneous external input $E(x) = E_c$ and are excited by their immediate neighbors and inhibited by neurons farther away. The weight matrix $w(x, x')$ is thus a function of the difference $x - x'$ only (see Figure 1A),

$$w(x, x') = k(x - x'),$$

and a typical weighting function or connectivity kernel $k(y)$ has the shape of a Mexican hat (see Figure 1B), for example,

$$k(y) = \frac{A}{\sigma_1} \exp\left(-\frac{y^2}{2\sigma_1^2}\right) - \frac{A}{\sigma_2} \exp\left(-\frac{y^2}{2\sigma_2^2}\right),$$

where σ_1 is smaller than σ_2.

This network can maintain stable bumps of activity at any position x (see Figure 1D). Intuitively, individual activity bumps are stable in the absence of inputs because the short-range excitatory connections provide sufficient

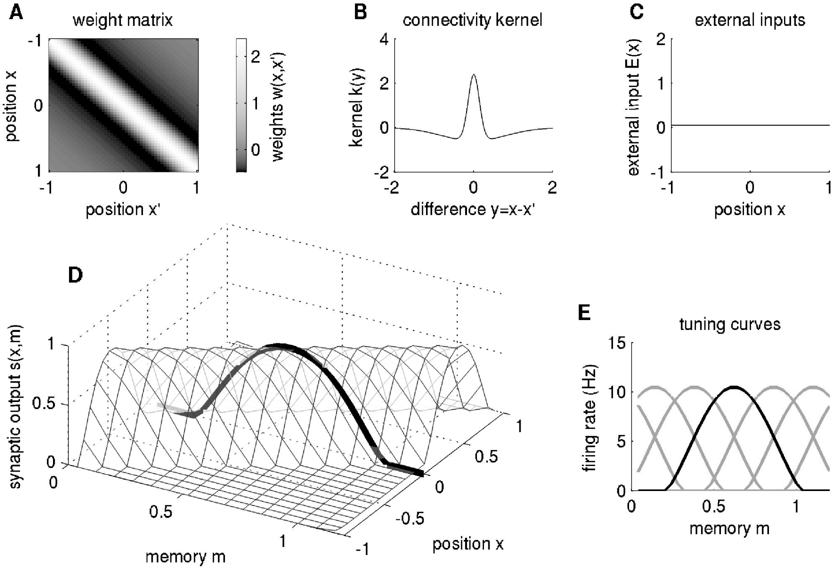

Figure 1: Bump attractor network. (A) The weight or connectivity matrix depends on only the difference $y = x - x'$ between the positions x and x' of two neurons. (B) The connectivity kernel $k(y)$ shows that nearby neurons excite each other, and neurons farther away inhibit each other. (C) All neurons receive the same constant external input. (D) The stationary activity profiles $s(x, m)$ are bell shaped, as a function of both x and m. The thick black curve shows the synaptic output of neuron $x = 0$ as a function of the memory m. (E) Tuning curves for neurons at different positions x. The thick black curve corresponds to the neuron at position $x = 0$.

excitatory feedback to maintain the neural activities within the bump. The long-range inhibitory connections counterbalance the uncontrolled growth of the activity bump. Since the network is infinitely large, this activity bump can be generated at any position x due to the translational symmetry of the network connectivity matrix.

For arbitrary input-output functions $f(\cdot)$, an analytical solution of the stationary network states cannot be obtained. Nonetheless, self-maintained activity bumps can exist in many networks, even if details of the connectivity kernel or the input-output function $f(\cdot)$ are changed or if the network is bounded (e.g., ranges from positions $x = -1$ to $x = 1$). As a function of the memories m, the tuning curves of individual neurons, equation 3.3, are bell shaped (see Figure 1E).

In its most common formulation, the neurons in a bump attractor are assumed to be organized on a ring, rather than a straight line, so that position $x = -1$ corresponds to position $x = 1$ (Ben-Yishai et al., 1995; Skaggs et al., 1995; Zhang, 1996). A possible weight matrix is given by (Ben-Yishai et al., 1995)—the connectivity kernel $k(y) = -J_0 + J_2 \cos(y)$. Note that this kernel has the additional property $k(y+1) = k(y-1)$ for $y = [-1, 1]$, which enforces the network's ring topology. Ring attractors have proven useful to explain the tuning curves found in systems that store information about angles, such as the head-direction system (Skaggs et al., 1995; Zhang, 1996; Taube & Bassett, 2003).

4 Line Attractor Design I: Translationally Invariant Activity Patterns

In both the bump and ring attractor model, the continuum of stable fixed points of the network is readily understood through the translational symmetry of the activity profile on the spatial axis. Similar geometric principles are not known for line attractors with monotonic tuning; instead, their construction has so far been based on numerical approximations of appropriate weight matrices (Seung, 1996; Seung et al., 2000; Miller et al., 2003; Eliasmith, 2005). We will now show that the class of solutions with this property—translational symmetry—can be significantly extended and, in particular, includes monotonic line attractors.

To investigate the class of translationally invariant solutions to equation 3.2, we set $s(x, m)$, the family of fixed-point solutions parameterized by m, to

$$s(x, m) = q(x - m),$$

with $x \in [-1, 1]$ and $m \in [m_0, m1]$, and observe that the following equation must hold:

$$\frac{\partial}{\partial x} q(x - m) = -\frac{\partial}{\partial m} q(x - m). \tag{4.1}$$

Inserting equation 3.2 into equation 4.1, we obtain

$$\begin{aligned} &f'(s_{\text{tot}}(x, m)) \frac{\partial}{\partial x}\left[\int_{-1}^{1} dx' w(x, x') q(x' - m) + E(x)\right] \\ &\quad = -f'(s_{\text{tot}}(x, m)) \frac{\partial}{\partial m}\left[\int_{-1}^{1} dx' w(x, x') q(x' - m) + E(x)\right]. \end{aligned}$$

This equation is trivially fulfilled for all areas in the (x, m)-plane in which $f'(s_{\text{tot}}(x, m)) = 0$, that is, areas where the synaptic outputs are either zero, $q(x - m) = 0$ or saturated, $q(x - m) = 1$. When $f'(s_{\text{tot}}(x, m)) \neq 0$ (corresponding to the area under the bump in Figure 1D), the synaptic inputs must obey the translation rules:

$$\frac{\partial}{\partial x}\left[\int_{-1}^{1} dx' w(x,x')q(x'-m)+E(x)\right]$$
$$=-\frac{\partial}{\partial m}\left[\int_{-1}^{1} dx' w(x,x')q(x'-m)+E(x)\right].$$

Using equation 4.1, we can reformulate the right-hand side to obtain

$$\frac{\partial}{\partial x}\left[\int_{-1}^{1} dx' w(x,x')q(x'-m)+E(x)\right]=\int_{-1}^{1} dx' w(x,x')\frac{\partial}{\partial x'}q(x'-m).$$

Using partial integration of the right-hand side and solving for $\partial_x E(x)$, we obtain

$$\frac{\partial}{\partial x}E(x)=w(x,1)q(1-m)-w(x,-1)q(-1-m)$$
$$-\int_{-1}^{1} dx'\left(\frac{\partial w(x,x')}{\partial x}+\frac{\partial w(x,x')}{\partial x'}\right)q(x'-m). \tag{4.2}$$

Equation 4.2 will be the central basis of our developments. Although it need hold only when $f'(s(x,m))\neq 0$, for simplicity we will confine ourselves here to solutions where equation 4.2 holds in the entire (x,m) plane. Below we describe some further simplifying assumptions that allow us to use equation 4.2 to design a wide variety of line attractors. Before we do so, however, we illustrate the simplest possible monotonic line attractor that we can design using equation 4.2.

5 A Basic Monotonic Line Attractor

Let us assume that the network has all-to-all excitatory connections with equal weights, $w(x,x')=w_E$ (see Figures 2A and 2B). According to equation 4.2, and noting that the derivatives of $w(x,x')$ vanish, the external inputs must fulfill the equation

$$\frac{\partial}{\partial x}E(x)=w_E(q(1-m)-q(-1-m)).$$

If we assume that the neuron at the left end, position $x=-1$, is inactive for all memories m, so that $s(-1)=q(-1-m)=0$, and the neuron at the right end, position $x=1$, is saturated so that $s(1)=q(1-m)=1$, then the external inputs must be (see Figure 2C)

$$E(x)=w_E x+E_c. \tag{5.1}$$

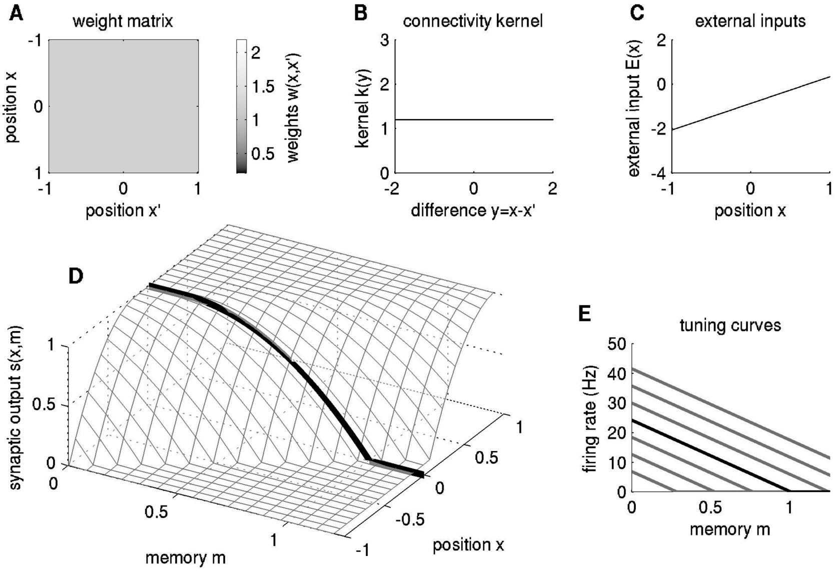

Figure 2: Monotonic line attractor network. (A) The network is all-to-all connected with equal excitatory weights, $w(x, x') = w_E$. (B) Accordingly, the connectivity kernel $k(y)$ is constant. (C) Neurons receive external inputs that depend linearly on their position x. (D) The stationary activity profiles $s(x, m)$ are sigmoidal functions of both x and m. The thick black curve shows the synaptic output of neuron $x = 0$ as a function of the memory m. (E) Tuning curves for neurons at different positions x. The thick black curve corresponds to the neuron at position $x = 0$. Although the synaptic outputs in panel D saturate, the firing rates of individual neurons do not.

Note that the external inputs must have this form if a translationally invariant steady-state $s(x, m) = q(x - m)$ is to exist. While a necessary condition, this choice of $E(x)$ guarantees neither that steady states exist nor that they are nontrivial: as an extreme example, equation 5.1 can be trivially satisfied with $w_E = 0$ and $E(x) = E_c$, but in that case, there is only a single stable state, and that is the trivial solution with all neurons having the same output, $s(x) = f(E_c)$.

For large enough w_E and a specific value of E_c, determined below, a continuum of different stable states does exist and is shown in Figure 2D. Notice that by design, neurons at the left end always have inputs below threshold to produce an output $s(-1) = 0$, while neurons at the right end always have their synaptic outputs saturated, $s(1) = 1$. Nonetheless, this last fact does not imply that neuronal firing rates at $x = 1$ must always be saturated, since $g(\cdot)$, the mapping of firing rate into synaptic output, may have reached saturation, even while $h(\cdot)$, the mapping of synaptic input into

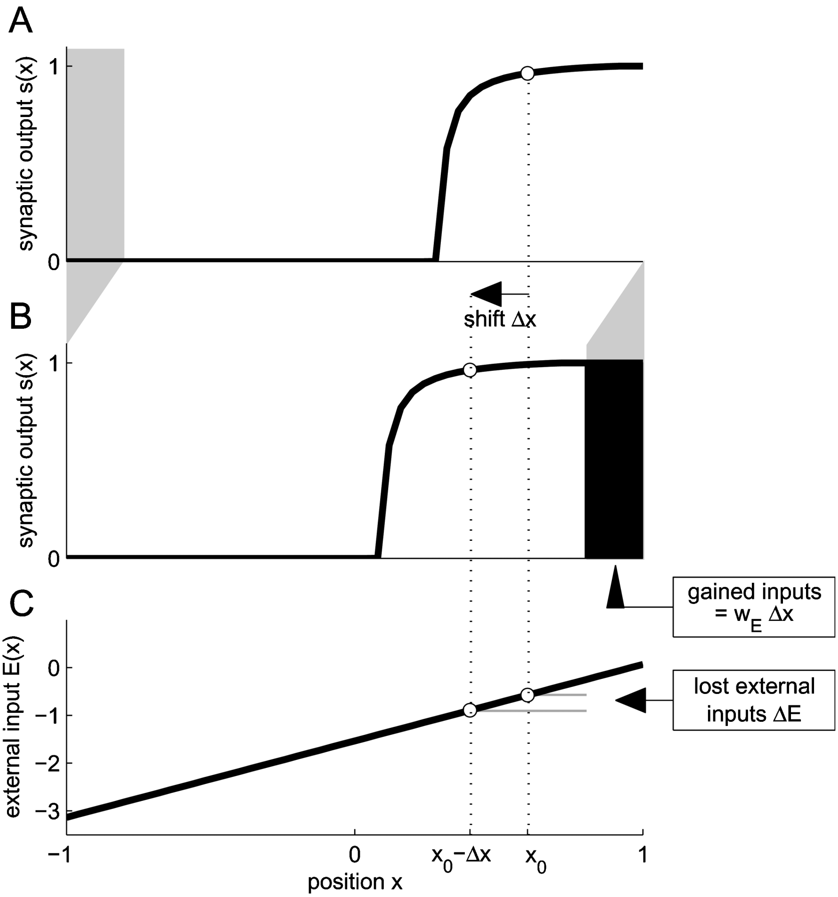

Figure 3: Design of a simple monotonic line attractor. (A) Stationary activity profile of the basic monotonic line attractor. (B) When we shift the network state by Δx to the left, the neuron at position $x_0 - \Delta x$ will have the same synaptic output that neuron x_0 had in panel A. However, the recurrent inputs to the neuron $x_0 - \Delta x$ are not the same, since the shift of the network state causes an overall increase in recurrent excitation (black area). (C) If the external inputs to neuron $x_0 - \Delta x$ exactly compensate for the gain in recurrent inputs, then neuron $x_0 - \Delta x$ in panel B will also be in steady state.

the firing rate, has not (compare Figures 2D and 2E and see also appendix A and Figure 6).

How does this network achieve its stability for a continuum of states? A particular stable state of the network is shown in Figure 3A. When the memory m is decreased, the activity pattern $s(x, m)$ is translated leftward, and the number of neurons in the network with saturated output increases (see Figure 3B). Consequently, the total recurrent excitation received by any neuron will also increase, leading to a gain in recurrent excitation that is proportional to the shift Δx as shown by the black area in Figure 3B. Now consider the neuron at position x_0 in Figure 3A. Let us denote this neuron's synaptic output as s_0. As m decreases and the activity pattern is translated toward the left, the value of x identifying the neuron with synaptic output s_0 decreases (see Figure 3B). For lower values of x, the external excitation $E(x)$ is smaller (see Figure 3C). When the gain in recurrent excitation matches this loss in external excitation, the forces on the neurons identified by $s(x, m) = s_0$ remain constant even when m changes. Equation 5.1 enforces this balance so that translations of any steady state obeying the boundary conditions will also be a steady state.

The intuition illustrated in Figure 3 aside, let us now determine the conditions on w_E and E_c that lead to nontrivial translationally invariant solutions. Define the integrated activity in the network as

$$u \equiv \int_{-1}^{1} dx\, s(x),$$

so that the steady state, equation 3.1, can be written as

$$s(x) = f(w_E u + w_E x + E_c).$$

To be consistent with the boundary conditions, the output $s(x)$ of the left-most ($x = -1$) neuron must be zero, and the output of the right-most neuron ($x = 1$) must be saturated at one. These conditions are reflected in two inequalities. For the left border neuron, the total synaptic input must be

$$s_{\text{tot}}(-1) = w_E u - w_E + E_c \leq s_{\text{th}}, \tag{5.2}$$

where s_{th} is the value of synaptic input below which there is zero output. Note that this equation must hold for all points on the line attractor—that is, for all memories m. These different points are distinguished through different values of the integrated activity u. Equality holds in equation 5.2 for the activity profile with the largest integrated activity, $u = u_L$. This is the state on the line attractor for which the activity of the left border neuron, $s(-1)$, is still zero, yet the activities of its neighbors are not (compare in Figure 2D, position $x = -1$, memory $m = 0$).

For the right border neuron (position $x = 1$), we obtain

$$s_{\text{tot}}(1) = w_E u + w_E + E_c \geq s_{\text{sat}}, \tag{5.3}$$

where s_{sat} is the synaptic input above which the neuron is in saturation. Equality holds for the steady state with the smallest integrated activity, $u = u_R$. Here, the right border neuron, $s(1)$, is still in saturation, yet its neighbors no longer are (compare in Figure 2D, position $x = 1$, memory $m = 1.25$). Using the cases $u = u_L$ in equation 5.2 and $u = u_R$ in equation 5.3, we can solve these equations for w_E to obtain

$$\begin{aligned} w_E &= \frac{s_{\text{sat}} - s_{\text{th}}}{2 + u_R - u_L} \\ &> \frac{1}{2}(s_{\text{sat}} - s_{\text{th}}), \end{aligned} \tag{5.4}$$

where the last inequality follows because $u_L > u_R$. Accordingly, only when w_E exceeds the right-hand side of equation 5.4 are we ensured that the left and right border neurons fulfill the boundary conditions. When $w_E = (s_{\text{sat}} - s_{\text{th}})/2$, then $u_R = u_L$, implying that there is only one stable point. If w_E is greater than $(s_{\text{sat}} - s_{\text{th}})/2$, then $u_L - u_R > 0$, and there is a range of u values that satisfies the boundary conditions. The maximum possible range, with $u_L - u_R = 2$, occurs when $w_E = \infty$. In a manner similar to the case of the bump attractor, the line attractor solutions require a large enough w_E but are independent of the detailed form of $f(\cdot)$.

These considerations leave only the parameter E_c unconstrained. In appendix B, we show that when w_E satisfies equation 5.4 and if E_c has the value

$$E_c = -w_E + s_{\text{sat}} - \int_{s_{\text{th}}}^{s_{\text{sat}}} ds' f(s'), \tag{5.5}$$

the network has a continuum of stationary states. Furthermore, in appendix E, we show that these stationary states are stable, so that a line attractor network is obtained.

6 Line Attractor Design II: Boundary Conditions

While a uniform connectivity matrix, $w(x', x) = w_E$, provides a particularly treatable simplification of equation 4.2, it is not the only one. The main issue to be addressed with respect to equation 4.2 is that $E(x)$ depends on both the weight matrix, $w(x, x')$, and the activity profiles, $q(x - m)$. However, by definition $E(x)$ cannot depend on m, so in order for the equation to be valid, the m-dependence on the right-hand side needs to drop out.

We will make two constraining assumptions that let m drop out of the right-hand side of equation 4.2. First, we will confine ourselves to solutions where equation 4.2 is valid for the entire (x, m) plane. Second, we further confine ourselves to weight matrices with Toeplitz symmetry, $w(x, x') = k(x - x')$. For these weight matrices, the integral on the right-hand side of equation 4.2 vanishes. Note that Toeplitz-type weight matrices are also used in the bump and ring attractor; here, however, we do not require the additional symmetry constraint, $w(x, x') = w(x', x)$, that is used in those models.

With these assumptions, we integrate equation 4.2 to obtain

$$E(x) = E_c + q(1-m)\int_{-1}^{x} dx' k(x'-1) - q(-1-m)\int_{-1}^{x} dx' k(x'+1). \tag{6.1}$$

While the right-hand side of this equation still depends on m, this dependence is now reduced to the activities of the border neurons—the neurons at the positions $x = -1$ and $x = 1$. There are only two possibilities to eliminate the remaining dependence on m and render a unique equation for $E(x)$:

- **Constant boundaries.** If the activity of the border neurons remains constant for all m, so that $q(-1-m) = \text{const1}$ and $q(1-m) = \text{const2}$, then the m-dependence drops out trivially. The simplest conditions are those for which the border neurons are either below threshold or in saturation. The monotonic line attractor designed above (see section 5) falls into this class, as does the bump attractor (see section 3). For the latter, both border neurons are below threshold.
- **Periodic boundaries.** If the connectivity kernel fulfills the additional constraint $k(x-1) = k(x+1)$ and if the boundaries are periodic, $q(-1-m) = q(1-m)$, then the m-dependence will drop out because the integrals in equation 6.1 cancel exactly. In this case, we have $E(x) = E_c$. This choice of external inputs, together with the condition $k(x-1) = k(x+1)$, does indeed create a network with ring topology since the neuron at position $x = -1$ and the neuron at position $x = 1$ become identical. (We note that Toeplitz matrices with this additional property are also called circulant matrices; Horn & Johnson, 1985.) The ring attractor discussed in section 3 falls into this class.

The choice of a weight matrix and a boundary condition therefore fully determines the x-dependence of the external inputs $E(x)$, up to an additive constant. Nonetheless, several uncertainties remain in the line attractor design since the above conditions are necessary but not sufficient. More specifically, for an arbitrary Toeplitz weight matrix and a choice of the boundary conditions, there is no guarantee that a value E_c exists such that a network with external inputs tuned according to equation 6.1 will have a nontrivial, stable activity profile consistent with the boundary conditions.

While we have not been able to specify all Toeplitz weight matrices that allow the design of line attractors, we outline several conditions below that allow constructing a large variety of line attractors (see also Figure 4). We will not consider the periodic boundary conditions or the boundary conditions $q(1-m) = q(-1-m) = 0$ since these correspond to the bump attractor and have already been discussed in great detail in the literature.

7 A Potpourri of Line Attractors Based on Toeplitz Symmetry

7.1 More Line Attractor Networks. A large set of line attractors can be designed with the boundary conditions $q(-1-m)=0$ and $q(1-m)=1$. All we need to do is follow the same principles as in section 5.

First, since not all connectivity kernels can support a line attractor with these boundary conditions, we need to find a condition that tells us which kernels do. As shown in appendix C, we can follow a similar line of thought as in the derivation of equation 5.4. If $k(y)$ fulfills the inequality

$$\int_{-1}^{1} dx \; \min\{k(1-x), k(-1-x)\} \geq s_{\text{sat}} - s_{\text{th}}, \tag{7.1}$$

then a line attractor network can be obtained by tuning E_c. Note that we retrieve equation 5.4 when $k(y) = w_E$. A simple way to obtain kernels that obey equation 7.1 is to make sure that excitation dominates the network connectivities. As a direct consequence, we note that noise correlations of simultaneously recorded neurons in such networks must be positive on average. This is indeed the case for simultaneously recorded neurons in the goldfish oculomotor system (Aksay, Baker, Seung, & Tank, 2003) or the monkey prefrontal cortex (Machens, Romo, & Brody, 2005), if the slope of the respective tuning curves has the same sign.

Second, we need to tune the constant E_c in equation 6.1. For arbitrary connectivity kernels that fulfill equation 7.1, this parameter has to be tuned numerically. In contrast to previous monotonic line attractor designs, numerical tuning has been reduced to the tuning of a single scalar parameter. Fortunately, as we now describe, a simple procedure to achieve this tuning can be devised. Let us assume that we know a steady-state $s_0(x)$ on the line attractor. For the correct value of E_c, all time derivatives, equation 2.1, will vanish if the network is in this steady state. If, on the other hand, E_c is too large, then the time derivatives will be positive at any position x, and if E_c is too small, then the time derivatives will be negative. Due to the monotonicity of $f(\cdot)$, the integral over the time derivatives,

$$D(E_c, s_0(\cdot)) = \int_{-1}^{1} dx \left[-s_0(x) + f\left(\int_{-1}^{1} dx' k(x-x') s_0(x') + E(x|E_c) \right) \right] \tag{7.2}$$

is therefore also a monotonic function of E_c, and its root yields the correct value for E_c. The notation $E(x|E_c)$ reminds us that the external inputs $E(x)$ depend on E_c, as defined in equation 6.1. The correct value for E_c can then be determined with simple root-finding methods (Press, Teukolsky, Vetterling, & Flannery, 1992).

The only remaining problem is that we do not know the precise form of $s_0(x)$ until we know E_c. In practice, though, if $s_0(x)$ is sufficiently close to an actual steady state, such as $s_0(x) = \Theta(x)$, the root of equation 7.2 will still lie in the neighborhood of the correct E_c. In this neighborhood, the dynamics of the network are usually slow along the line that will give rise to the line attractor and fast along all other directions. Consequently, if we simulate the network differential, equation 2.1, starting with a state that is sufficiently close to the actual line attractor, such as $s_0(x) = \Theta(x)$, the shape of $s_0(x)$ will rapidly approximate the shape of the steady-state activity profiles and only slowly drift along the line attractor. By determining the root of equation 7.2 at every time step, we can adjust E_c during the simulation. While the network dynamics falls into the line attractor, E_c will then converge onto its correct value (see also appendix A).

Several examples designed with this recipe are shown in Figures 4A to 4F. Whereas most of these line attractors have monotonic tuning curves, it is also possible to obtain nonmonotonic tuning curves, as shown in Figure 4F.

7.2 Positive and Negative Monotonic Tuning Curves I: Revisiting the Bump Attractor. Neurons in the networks shown in Figures 4A to 4E have tuning curves with positive monotonic slopes. In electrophysiological recordings from the prefrontal cortex, however, one finds both neurons with positive and neurons with negative monotonic tuning curves within the same system (Romo et al., 1999; Machens et al., 2005). We will show next that both types of tuning curves can be obtained by choosing a network with boundary conditions $q(-1-m) = 1$ and $q(1-m) = 1$.

As previously, the general problem is to find weight matrices that match this particular choice of saturated synaptic output boundary conditions. Here, our approach is heuristic: we choose weight matrices such that the tuning of the external inputs $E(x)$, through equation 6.1, creates external inputs at the boundaries, $E(x = -1)$ and $E(x = 1)$, that support the chosen boundary conditions. In other words, to guarantee nontrivial steady states, we seek a U-shaped function $E(x)$ such that neurons close to the borders receive large external inputs and neurons in the center (positions around $x = 0$) receive small external inputs. According to equation 6.1, a simple way to obtain this shape of the external inputs is to choose a Toeplitz weight matrix such that the kernel $k(y)$ is negative around its ends ($y \approx -2$ and $y \approx 2$) and positive in the center ($y \approx 0$).

An example of such a line attractor is shown in Figure 4I. Note that the weight matrix of this network is identical to the one for the bump attractor network in Figure 4H. Nonetheless, simply by changing the external inputs, without changing the internal recurrent connectivity at all, it is possible to transform a bump attractor into a line attractor with monotonic tuning curves. Individual neurons are tuned to the memory m with both positive and negative monotonic slopes.

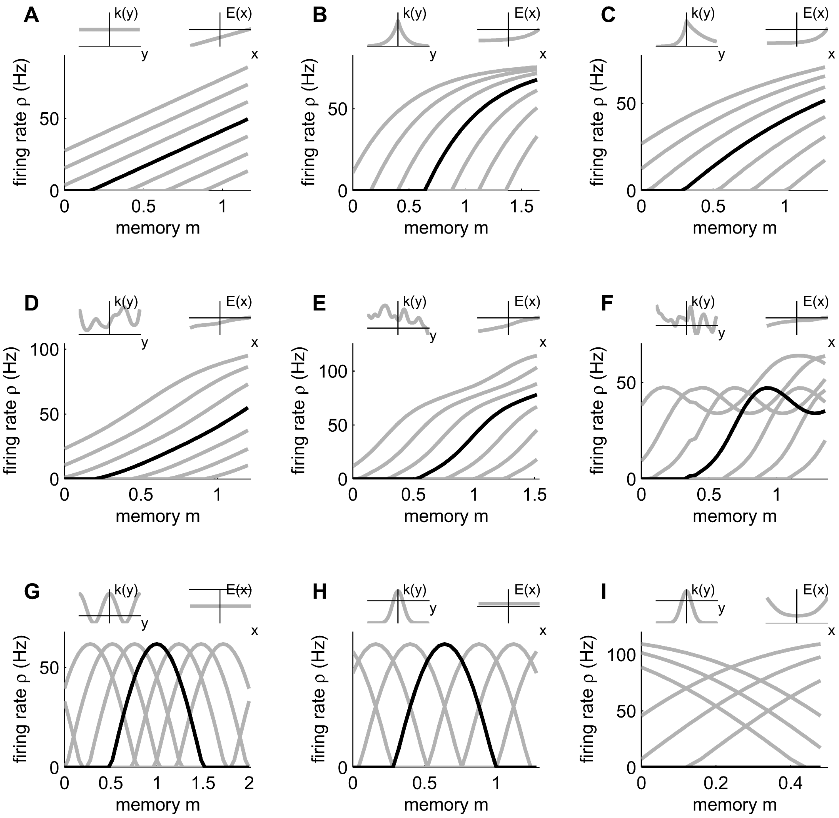

Figure 4: Tuning curves obtained from line attractor networks with different kernels $k(y)$ or different boundary conditions. The thick black lines correspond to the tuning curve of the neuron at position $x = 0$. The insets show the connectivity kernels $k(y)$ and the external inputs $E(x)$. (A) Line attractor with $k(x - x') = w_E$ and boundary conditions $s(-1) = 1$ and $s(1) = 0$. This is the basic monotonic line attractor (see section 5) with inverted boundary conditions and tuning curves. (B) Line attractor with a symmetric connectivity kernel and short-range excitatory connections. The boundary conditions are again $s(-1) = 1$ and $s(1) = 0$. (C) Line attractor with the same boundary conditions but an asymmetric kernel. (D–F) Line attractors with the same boundary conditions but randomized connectivity kernels. Note that in all of these kernels, excitatory connections dominate. Panel F shows that nonmonotonic tuning curves are possible as well. (G) Ring attractor (see section 3). (H) Variant of the bump attractor with short-range excitation and global inhibition. Boundary conditions are $s(-1) = 0$ and $s(1) = 0$. (I) By changing the external inputs to the network in H (boundary conditions are now $s(-1) = 1$ and $s(1) = 1$), we can obtain a network with both positive and negative monotonic tuning curves. Here neurons in the center (black line; position $x = 0$) are not active.

We remark on this line attractor simply to demonstrate the range of possibilities obtainable by tuning the external inputs. However, we have not investigated this family of line attractors in greater detail. In the example in Figure 4G, the parameter E_c was tuned by trial and error; the exact parameter values for this network are given in appendix A.

7.3 Positive and Negative Monotonic Tuning Curves II: Coupling Line Attractor Networks. The definition of m is purely conventional. By simply redefining it, that is, by setting $m \mapsto -m$, or, equivalently, setting $x \mapsto -x$, the networks in Figures 4A to 4E will have tuning curves with negative slopes. In principle, one could therefore obtain opposite tuning curves by constructing two independent networks. However, two independent networks could lead to conflicting representations of the analog memory m, which is, after all, one-dimensional. To solve this problem, we need to ask how to couple these networks so that both represent the same m. These considerations will provide a different solution to the problem of constructing networks with both positive and negative monotonic tuning curves. Since the question of coupling line attractor networks is interesting in its own right, we discuss it next.

Imagine that we have two networks with Toeplitz weight matrices $w_{11}(x, x') = k_{11}(x - x')$ and $w_{21}(x, x') = k_{22}(x - x')$. To couple these networks, we choose Toeplitz coupling matrices, so that the connections from the first to the second network are given by $w_{12}(x, x') = k_{12}(x - x')$ and the reverse connections by $w_{21}(x, x') = k_{21}(x - x')$. The steady-state equations for the two layers are simply rewritten from equation 3.2,

$$s_i(x) = f\left(\sum_{j=1}^{2} \int_{-1}^{1} dx' k_{ij}(x - x') s_j(x') + E_i(x)\right),$$

where $s_i(x)$ is the activity at position x in the ith layer with $i \in \{1, 2\}$. A derivation similar to the one in sections 4 and 6 shows that the external inputs are again determined by the weight matrix and the boundary conditions,

$$\begin{aligned} E_i(x) = E_{i,c} + \sum_{j=1}^{2} \Bigg[& q_j(1 - m) \int_{-1}^{x} dx' k_{ij}(x' - 1) \\ & - q_j(-1 - m) \int_{-1}^{x} dx' k_{ij}(x' + 1) \Bigg]. \end{aligned}$$

As discussed before, the boundary conditions need to be self-consistent with the Toeplitz weight matrices. Due to the increase in possibilities, there are far more free parameters in this problem than before, which makes a self-consistent design more complicated.

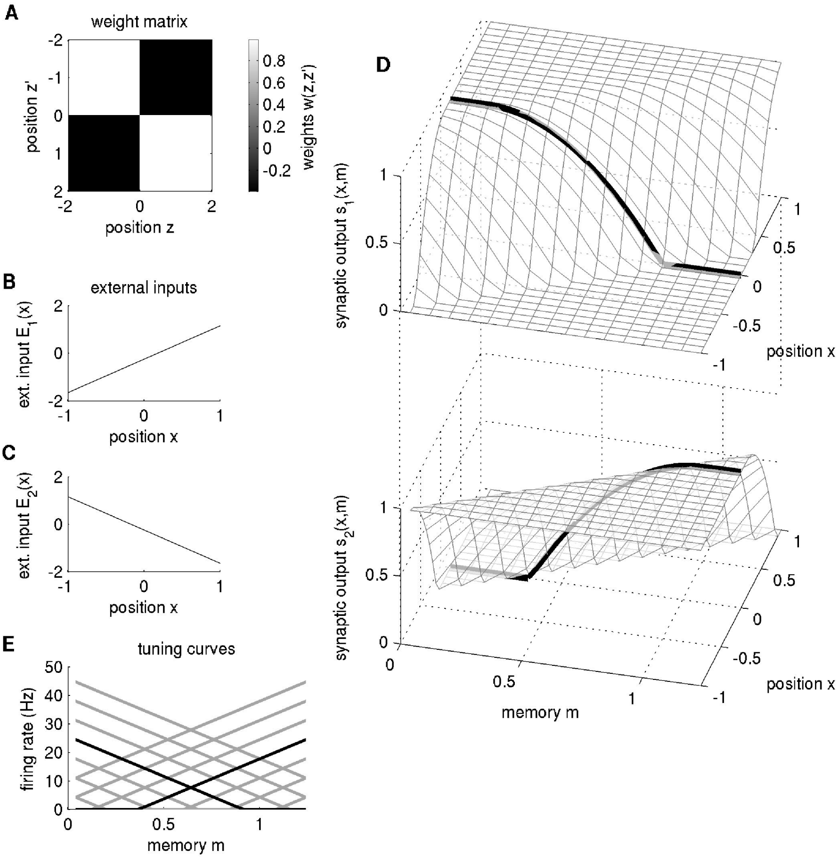

Figure 5: Coupling of two line attractor networks. (A) Connectivity matrix of the coupled network. Within each layer, connections are excitatory with equal weights; between layers, they are inhibitory with equal weights. (B, C) Neurons in the two layers receive external inputs that increase (layer 1) or decrease (layer 2) as a function of their position. (D) Activity profiles $s_1(x, m)$ and $s_2(x, m)$ in the two layers. Note the opposite tuning with respect to both x and m. The thick black curve shows the synaptic output of neurons at position $x = 0$ as a function of the memory m. (E) Tuning curves for neurons at different positions x. The thick black curves correspond to the neurons at position $x = 0$.

A simple case is given by building on the example of section 5. Let us take two of these networks, so that $k_{11}(x - x') = k_{22}(x - x') = w_E$, and couple them with constant, mutually identical connections, $k_{12}(x - x') = k_{21}(x - x') = w_I$. While these connections could be either excitatory or inhibitory, the index I insinuates that we will eventually assume that they are inhibitory (see also Figure 5A).

We will assume that the first layer has negative monotonic tuning curves (achieved by setting $s_1(-1) = q_1(-1-m) = 0$ and $s_1(1) = q_1(1-m) = 1$) and the second layer positive monotonic tuning curves (achieved by setting $s_1(-1) = q_1(-1-m) = 1$ and $s_1(1) = q_1(1-m) = 0$). The external inputs therefore become (compare equation 5.1)

$$E_1(x) = (w_E - w_I)x + E_c$$
$$E_2(x) = -(w_E - w_I)x + E_c,$$

leading to the steady-state equations

$$s_1(x) = f(w_E u_1 + w_I u_2 + (w_E - w_I)x + E_c) \tag{7.3}$$
$$s_2(x) = f(w_E u_2 + w_I u_1 - (w_E - w_I)x + E_c)$$

with $u_1 = \int dx' s_1(x')$ and $u_2 = \int dx' s_2(x')$. With similar methods as in the one-layer case, we can derive the parameter regime for which this network yields a line attractor. An outline of the derivations is given in appendix D. For the parameters of the weight matrix, we find that

$$w_E - w_I \geq \frac{1}{2}(s_{\text{sat}} - s_{\text{th}}). \tag{7.4}$$

Note that if w_I is inhibitory, its value is negative and can therefore be chosen arbitrarily. Furthermore, we find that E_c can take a whole range of values:

$$E_c \geq -\frac{2w_I}{w_E - w_I} s_{\text{sat}} + w_I - w_E - \frac{w_E + w_I}{w_E - w_I}\left[\int_{s_{\text{th}}}^{s_{\text{sat}}} ds' f(s') - s_{\text{sat}}\right] \tag{7.5}$$

$$E_c \leq -\frac{2w_I}{w_E - w_I} s_{\text{th}} - 3w_I - w_E - \frac{w_E + w_I}{w_E - w_I}\left[\int_{s_{\text{th}}}^{s_{\text{sat}}} ds' f(s') - s_{\text{sat}}\right]. \tag{7.6}$$

The distribution of tuning curves is shown in Figure 5E. Note that the overall network activity $U = u_1 + u_2$ is constant. The memorized variable is determined by $m = -u_1 = U - u_2$.

We note two additional aspects about the coupled network. First, since the overall network activity is constant, the external, x-dependent inputs could also be supplied through the network itself, that is, made internal by appropriate changes of the weight matrix. If $w(z, z')$ denotes the weight matrix of the complete network (see Figure 5A), then we can write $w'(z, z') = w(z, z') + c(z)$ and $E'(z) = E(z) - Uc(z)$ for an arbitrary function $c(z)$ and the network with $w'(x, x')$ and $E'(x)$ will have a line attractor as well.[1]

[1] Quite generally, if there exists a function $h(x, x')$ with $\int dx' h(x, x') s(x', m) = c(x)$ for all indices m on the line attractor, then this indicates an invariance of the weight matrix and

Second, we note that by concatenating the two layers such that position $x = 1$ in layer one corresponds to position $x = -1$ in layer two, we obtain a single layered network with a single shiftable activity profile. This illustrates that the existence of (single) shiftable activity profiles is not restricted to pure Toeplitz-type weight matrices. The restriction to these weight matrices was a simplification introduced for the derivation of equation 6.1.

8 From Continuous to Discrete Networks

All of our derivations so far have been done for a continuous layer of neurons. While this has the benefit of simplifying the analysis, in both reality and simulations, only a finite number of neurons will be available. In these discrete networks, we can still fine-tune the external inputs, and the exact, discrete analog to equation 6.1 is provided in appendix A. The finite networks then exhibit a discrete series of fixed points rather than a continuous line, since the activity profile can be shifted only one neuron at a time. The construction of line attractor networks based on translational symmetry is therefore not restricted to large-scale networks whose dynamics are well described by the neural field equations (see equation 2.1) but can be employed in smaller networks as well. For instance, all the networks in Figure 4 were simulated with $N = 51$ units.

9 Discussion

The investigation of neural networks that give rise to persistent activity has a long history (Wang, 2001; Major & Tank, 2004). Here, we have focused on graded persistent activity, which is thought to be the neural substrate of the short-term memory of analog values. The design of networks that exhibit graded persistent activity is relatively straightforward if the neuron's input-output functions are linear or threshold linear (Morishita & Yajima, 1972; Cannon, Robinson, & Shamma, 1983; Seung, 1996); however, the construction of such networks has proven more complicated for nonlinear, realistic input-output functions and has often relied on numerical tuning (Droulez & Berthoz, 1991; Zipser, Brandt, Littlewort, & Fuster, 1993; Seung, 1996; Seung et al., 2000; Eliasmith, 2005).

The bump attractor has been a prominent exception and has shown that the emergence of a line attractor can be tied back to an underlying symmetry in the network connectivity (Amari, 1977; Ben-Yishai et al., 1995; Skaggs et al., 1995). In this work, we have shown that this symmetry principle can be used to design a plethora of line attractor networks if one allows external inputs that are inhomogeneous. While symmetry may not be a necessity for

external inputs, and we can set $w'(x, x') = w(x, x') + \alpha h(x, x')$ and $E'(x) = E(x) - \alpha c(x)$ for any number α without changing the stationary states of the network.

the existence of continuous attractors—after all, numerical approximations of continuous attractor networks usually do not make that assumption—it provides additional insights and allows some analytical treatment of the respective networks.

Perhaps most important, the geometric understanding of symmetry-based line attractors put forth here may provide a useful conceptual basis for solving a major outstanding problem: What plasticity rules does biology use in constructing line attractors? To date, models of line attractors have not addressed how they might be constructed in biology. Line attractors produced through numerical tuning (Seung et al., 2000; Eliasmith, 2005; Singh & Eliasmith, 2006, for example) typically have a complex network connectivity that does not easily lend itself to devising biologically plausible plasticity rules with which they could be constructed. We speculate and hope that the far simpler underlying structure of the translational symmetry-based line attractors may facilitate such a development and generalize from the artificial models constructed here to biological line attractors.

In our framework, we assume a Toeplitz-symmetric weight matrix and then tune the external inputs so that (possible) stationary states of the network can be translated along the network's spatial axis. The continuous attractor therefore emerges due to the shift symmetry that the tuned external inputs create.[2] A crucial ingredient in creating this symmetry for (monotonic) line attractors is the saturation of synaptic outputs (see also Seung et al., 2000).

A prominent feature of all of our networks is that the tuning curves of different neurons are shifted along the axis representing the memorized variable, the m-axis. These shifts are often only partially visible (see Figure 4), as should be expected given the finite and bounded nature of both the memory m and the network itself. Many tuning curves found in neural systems, including the ones that are monotonic, show this kind of shifting and are therefore at least consistent with the idea of continuous attractors built on translational invariance. It should be mentioned, however, that tuning curves in real systems are usually a lot more heterogeneous than the ones we find in our simplified models. For instance, the slopes of monotonic tuning curves vary in both the oculomotor system of the goldfish (Seung et al., 2000) and the prefrontal cortex (Brody et al., 2003). While such heterogeneity could in principle be built into our line attractor networks (see also appendix E), it would simply be an add-on without any explanatory value. For simplicity, we have therefore neglected this issue. (For models that are explicitly based on such heterogeneity, see Eliasmith, 2005.)

For analytical simplicity, we have derived our results within the mean field framework and assumed that all synapses are identical except for their

[2] Indeed, this principle applies to both stable and unstable fixed points of the dynamics. When inverting the spatial axis in one of the layers of the coupled network (see Figure 5), one can obtain a continuous line of unstable fixed points as well (simulations not shown).

(positive or negative) weight. When we distinguish explicitly between different synapse types, such as excitatory and inhibitory synapses, we can still design a continuum of stationary states by fine-tuning the external inputs to these networks along the lines of equation 6.1. However, there exist circumstances in which the different time constants of excitation and inhibition can lead to dynamical instabilities (Wang, 1999; Pinto & Ermentrout, 2001), rendering the stationary points unstable. While assuming slow excitatory dynamics, for example, through NMDA channels, alleviates this problem (Wang, 1999), future work will have to determine the precise conditions under which the continuous attractor networks presented here are stable.

Last, but not least, all of our networks require that synaptic weights and external inputs are precisely tuned to the right parameters. This is a generic problem that all line attractor networks face, whether they are produced through symmetry principles or through numerical tuning, and it is an issue that this work is not meant to address. We want to add, though, that all of the networks here can be made robust against small changes in their parameters if we assume that individual neurons are hysteretic as originally proposed by Koulakov et al. (2002). Indeed, a spiking neuron implementation of a robust version of the coupled line attractor network (see Figure 5) has been described in Machens et al. (2005).

Appendix A: Simulations

In this section, we briefly describe the details necessary to replicate the simulations. In particular, we provide all the parameters for the networks shown in Figure 4.

A.1 Input-Output Function.

The construction of the synaptic input-output function $s = f(s_{\text{tot}})$ from the mapping of firing rates into synaptic outputs, $s = g(r)$, and the mapping of synaptic inputs to firing rates, $r = h(s_{\text{tot}})$, is shown in Figure 6. For the construction of line attractor networks along the lines presented here, the detailed shape of the input-output function $s = f(s_{\text{tot}})$ does not matter, as long as it is monotonic and as long as $f(s_{\text{tot}}) = 0$ if $s_{\text{tot}} \leq s_{\text{th}}$ and $f(s_{\text{tot}}) = 1$ if $s_{\text{tot}} \geq s_{\text{sat}}$ (compare Figure 6C).

In our network models, we assumed that synapses will saturate for sufficiently high presynaptic firing rates. One prominent form of synaptic saturation is given through synaptic depression. If the probability of vesicle release recovers with τ, then the postsynaptic conductance g will follow the equation (Dayan & Abbott, 2001),

$$g = g_{\max} \frac{r}{1 + (1 - f_D)\tau r}$$

where $g_{\max}$ is the maximum conductance input and f_D controls the strength of depression. Note that the precise biophysical mechanism for saturation

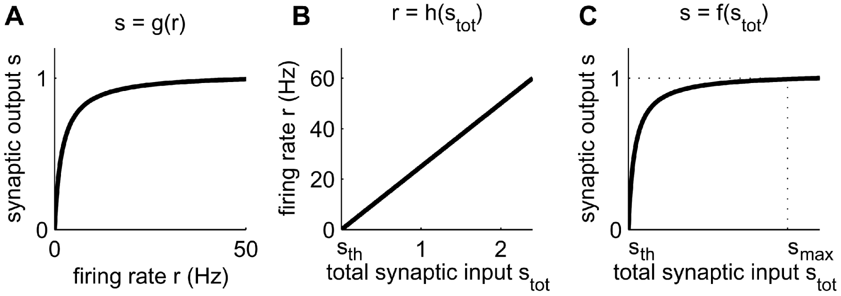

Figure 6: A neuron's input-output function. (A) The synaptic output of a neuron as a function of its firing rate. The synaptic output is proportional to the conductance input to the postsynaptic neuron. In this example, it saturates through synaptic depression. (B) Firing rate of a neuron as a function of its total synaptic input. (C) Combining A and B allows plotting a neuron's synaptic output as a function of its input; we call this function the neuron's synaptic input-output (i/o) function.

does not matter in our framework. Previous work has shown that NMDA synapses yield a more robust implementation of synaptic saturation in biophysically realistic networks (Wang, 1999).

Based on the above equation, we formulate the relation between the presynaptic firing rate r and the synaptic output s. Choosing $f_D = 0.5$ and $\tau = 1$ sec, we assume that the synaptic output of a neuron obeys the relation

$$s = g(r) = \begin{cases} 0 & \text{if } r \leq 0 \text{ Hz} \\ A\frac{r}{1+r/2} & \text{if } 0 < r < 50 \text{ Hz} \\ 1 & \text{if } r > 50 \text{ Hz}, \end{cases}$$

where $A = 13/25$ is chosen to make this function continuous. The piecewise definition of the function serves to ensure that the above-saturation regime is exactly, and not just approximately, constant.

For simplicity, we assume a threshold-linear dependency of the firing rate on the synaptic input,

$$r = h(s_{\text{tot}}) = \begin{cases} 0 & \text{if } s_{\text{tot}} \leq 0 \\ 50 s_{\text{tot}} & \text{if } s_{\text{tot}} > 0. \end{cases}$$

For the simulations in Figure 4, s_{tot} varies approximately between 0 and 2, leading to firing rates r that range from 0 to 100 Hz.

The input-output function is then given by $s = f(s_{\text{tot}}) = g(h(s_{\text{tot}}))$, as shown in Figure 6.

A.2 Discrete Networks. While our derivations have been done for continuous mean field networks, the simulations in all of the figures were performed with relatively small discrete networks (network size $N = 50, \ldots, 500$ neurons). When rewriting the main equations for finite networks, a few subtleties need to be taken into account.

For a network with N neurons, we replace the continuous position x by a discrete index $i = 1, \ldots, N$ and rewrite equation 3.1:

$$s_i = f\left(\sum_{j=1}^{N} w_{ij} s_j + E_i\right).$$

The derivation in section 4 can be performed for a discrete network as well, resulting in the following analog to equation 6.1,

$$E_i = E_c + s_1 \sum_{j=2}^{i} w_{j-1,N} - s_N \sum_{j=2}^{i} w_{j,1},$$

where it is implicitly understood that the weight matrix w_{ij} has a Toeplitz structure such that $w_{ij} = k_{i-j}$. Note the index shift in j between the second and third sums. Also note that the definition of E_c here is different from the one in equation 6.1 since the indices i are not symmetric around zero. (The position x is symmetric around zero.)

To calculate the value of E_c for the networks in Figures 4A to 4F, we simulate the following set of (differential) equations:

$$\dot{s}_i = -s_i + f\left(\sum_{j=1}^{N} w_{ij} s_j + E_i\right)$$

$$0 = \sum_{i=1}^{N}\left[f\left(\sum_{j=1}^{N} w_{ij} s_l + E_j(E_c)\right) - s_i\right],$$

with the initial conditions $s_i = 0$ for $i < N/2$ and $s_i = 1$ otherwise. The second equation is an implicit equation for E_c that can be solved at every time step using standard numerical root-finding methods (Press et al., 1992). While the first equation will evolve onto a continuous attractor, the solution of the second equation will converge onto the correct value of E_c.

The values of E_c in the networks in Figures 4G to 4I were estimated with a simple trial-and-error procedure and are given below.

Table 1: Design of Continuous Attractor Networks.

4	w_{ij}	s_1	s_N	E_c
Figure 4A	$1/25$	1	0	-1.924
Figure 4B	$3/25\exp(-\|i-j\|/12)$	1	0	-1.308
Figure 4C	$3/50\Theta(i-j)\exp(-\|i-j\|/30)$			
	$+3/50\Theta(-(i-j))\exp(-\|i-j\|/8)$	1	0	-0.4
Figure 4G	$1/25+2/25\cos(2\pi(i-j)/N)$	0	0	$-1 \in [-1.13, -0.7]$
Figure 4H	$-7/50+1/5\exp(-(i-j)^2/160)$	0	0	$0.3 \in [-0.19, 4.5]$
Figure 4I	$-7/50+1/5\exp(-(i-j)^2/160)$	1	1	$3 \in [1.5, 5.13]$

A.3 Parameters for Line Attractors in Figure 4. For the line attractors in Figure 4, we used networks with $N = 51$ neurons. The parameters of the weight matrices w_{ij} and the constant external inputs E_c are given in Table 1. The intervals in the last column indicate that the respective continuous attractor systems exist for several values of E_c.

Appendix B: Basic Monotonic Line Attractor: Derivation of E_c

To determine the correct value of E_c for the basic monotonic line attractor, we note that the dynamics of the network will always reduce to a single dimension, since the weight matrix has rank one. This statement is proved in appendix E. The relevant variable for the basic monotonic line attractor is the integrated network activity u. Using equation E.2, we obtain:

$$
\begin{aligned}
u &= \int_{-1}^{1} dx f(w_E u + w_E x + E_c) \\
&= \frac{1}{w_E}[F(w_E u + w_E + E_c) - F(w_E u - w_E + E_c)],
\end{aligned}
\tag{B.1}
$$

where $F(\cdot)$ is the antiderivative of $f(\cdot)$. Without loss of generality, we can choose $F(\cdot)$ such that $F(s_{\mathrm{th}}) = 0$. This function is then piece-wise defined as

$$
F(s_{\mathrm{tot}}) = \begin{cases} 0 & \text{if } s_{\mathrm{tot}} \le s_{\mathrm{th}} \\ \displaystyle\int_{s_{\mathrm{th}}}^{s_{\mathrm{tot}}} ds' f(s') & \text{if } s_{\mathrm{th}} < s_{\mathrm{tot}} \le s_{\mathrm{sat}} \\ F(s_{\mathrm{sat}}) + s_{\mathrm{tot}} - s_{\mathrm{sat}} & \text{if } s_{\mathrm{tot}} > s_{\mathrm{sat}}. \end{cases}
\tag{B.2}
$$

We note that the right term on the right-hand side of equation B.1 will vanish because of equation 5.2. Furthermore, the input to $F(\cdot)$ in the left term on the right-hand side of equation B.1 will always be larger than s_{sat}, which follows from equation 5.3. Using the definition of $F(\cdot)$ in equation B.2, we therefore obtain for equation B.1,

$$u = \frac{1}{w_E}[F(s_{\text{sat}}) + w_E u + w_E + E_c - s_{\text{sat}}],$$

which will have a continuum of solutions for u iff

$$E_c = -w_E + s_{\text{sat}} - F(s_{\text{sat}}),$$

which proves equation 5.5.

Appendix C: Line Attractors with One Saturated Boundary: Existence Proof and Constraints

C.1 Constraints on Connectivity Kernel. Given the boundary conditions $s(-1) = 0$ and $s(1) = 1$ and the appropriately tuned external inputs $E(x)$, equation 6.1, we can formulate inequalities, similar to equations 5.2 and 5.3, that need to be fulfilled for the left-most and right-most border neurons:

$$\int_{-1}^{1} dx\, k(-1-x)s(x) + E_c \geq s_{\text{th}} \tag{C.1}$$

$$\int_{-1}^{1} dx\, k(1-x)s(x) + \int_{-1}^{1} dx\, k(x-1) + E_c \leq s_{\text{sat}}. \tag{C.2}$$

Solving these inequalities for E_c and combining them leads to a single inequality:

$$\int_{-1}^{1} dx\, [k(1-x) - k(-1-x)]s(x) + \int_{-1}^{1} dx\, k(x-1) \geq s_{\text{sat}} - s_{\text{th}}. \tag{C.3}$$

Since we do not know the stationary network states $s(x)$ for arbitrary kernels $k(y)$, we can only approximate the left-hand side. If we assume that $s(x)$ takes a form that leads to the smallest possible value of the left integral, then this approximation will be conservative. Let us define the set of all x for which the term in the square brackets is negative: $\Omega = \{x | k(1-x) < k(-1-x), x \in [-1, 1]\}$. To minimize the left integral with respect to $s(x)$, we assume that $s(x) = 1$, whenever $x \in \Omega$ and $s(x) = 0$ otherwise. Accordingly,

$$\int_{\Omega} dx\, [k(1-x) - k(-1-x)] + \int_{-1}^{1} dx\, k(x-1) \geq s_{\text{sat}} - s_{\text{th}}.$$

Simplifying, we obtain

$$\int_{\Omega} dx\, k(1-x) + \int_{[-1,1]\backslash\Omega} dx\, k(-1-x) \geq s_{\text{sat}} - s_{\text{th}},$$

which is identical to equation 7.1. Note that this is a sufficient but not necessary condition on the kernels $k(y)$. In other words, there may be kernels that do not fulfill this inequality yet still allow the construction of a line attractor with the required boundary conditions.

C.2 Existence of Stationary States. To show that a line attractor exists for kernels that fulfill equation 7.1, we need to prove that—for at least one value E_c—the network has a stable stationary state that obeys the boundary conditions. Due to the tuning of the external inputs in equation 6.1, we know that if one such stationary state exists, it can be shifted within a certain range, and a continuum of stationary states must exist as well.

Given that a line attractor exists, we observe that the equality sign in equation C.1 will hold for the left-most state on the line attractor, which we denote as $s_L(x)$. This suggests that the correct choice for E_c is

$$E_c = s_{\text{th}} - \int_{-1}^{1} dx\, k(-1-x) s_L(x). \tag{C.4}$$

While this choice of E_c is motivated by the left-most state on the putative line attractor, we will next assume that $s_L(x)$ could be any activity profile. In turn, we can investigate the properties of states $s_L(x)$ that are solutions to the steady-state equation, equation 3.1. Given equations 6.1 and C.4, these must fulfill

$$s_L(x) = f\left(\int_{-1}^{1} dx' [k(x-x') - k(-1-x')] s_L(x') + s_{\text{th}} + \int_{-1}^{x} dx'\, k(x'-1) \right).$$

Certainly this equation will have at least one solution for $s_L(x)$. While we cannot specify this solution exactly, we can investigate its properties at the boundaries $x=-1$ and $x=1$. Plugging in the value $x=-1$, we notice that $s_L(-1)=0$, so that the solution fulfills the left boundary condition. Plugging in $x=1$, we notice from equation C.3 that the synaptic inputs are in saturation, so that $s_L(1)=1$. Accordingly, the choice of E_c in equation C.4 guarantees the existence of a stationary state that fulfills the required boundary conditions.

To determine the stability of these stationary states is difficult as long as their precise form is not known. Practical experience suggests, though, that they are always stable, thus giving rise to a line attractor.

Appendix D: Derivation of Parameter Regime for Mutual Inhibition Network

For the mutual inhibition network, the sum of the integrated activities of the first and second layer is a constant (compare Figure 5),

$$U = u_1 + u_2 = \text{const},$$

so that the equation for u_1 suffices to describe the system. Similar to the one-layer case, discussed in appendix B, we obtain an equation for u_1 by integrating the stationary network, equation 7.3, to obtain

$$\begin{aligned} u_1 = \frac{1}{w_E - w_I}&[F((w_E - w_I)(u_1 + 1) + w_I U + E_c) \\ &- F((w_E - w_I)(u_1 - 1) + w_I U + E_c)], \end{aligned} \tag{D.1}$$

where $F(\cdot)$ is the antiderivative of $f(\cdot)$ as given in equation B.2. Looking at the boundaries in the first layer, we find that for any state u_1, the following conditions must hold:

$$s_{\text{tot},1}(-1) = (w_E - w_I)(u_1 - 1) + w_I U + E_c \leq s_{\text{th}} \tag{D.2}$$

$$s_{\text{tot},1}(1) = (w_E - w_I)(u_1 + 1) + w_I U + E_c \geq s_{\text{sat}}, \tag{D.3}$$

and similar conditions hold for the opposite layer. Subtracting equation D.2 from equation D.3 yields the condition on the weights $w_E - w_I$ given in equation 7.4. The two inequalities also show that the first term in equation D.1 will always be in saturation, and the second term will always be below threshold. Using equation B.2, we obtain

$$u_1 = u_1 + \frac{1}{w_E - w_I}[F(s_{\text{sat}}) - s_{\text{sat}} + w_I U + (w_E - w_I) + E_c],$$

which will be valid if the second term vanishes. This results in the following equation for the summed activity U:

$$U = -\frac{1}{w_I}(F(s_{\text{sat}}) - s_{\text{sat}} + w_E - w_I + E_c). \tag{D.4}$$

Following the arguments for the one-layered case (see section 5), there are two possibilities for a left-most state. Either the activity profiles run into the threshold condition in the first layer, in which case,

$$\begin{aligned} s_{\text{tot},1}(-1) &= (w_E - w_I)(u_L - 1) + w_I U + E_c = s_{\text{th}} \\ s_{\text{tot},2}(-1) &= (w_E - w_I)(u_R + 1) + w_I U + E_c \geq s_{\text{sat}}, \end{aligned} \tag{D.5}$$

or the left-most state runs into saturation condition in the second layer, in which case,

$$\begin{aligned} s_{\mathrm{tot},1}(-1) &= (w_E - w_I)(u_L - 1) + w_I U + E_c \leq s_{\mathrm{th}} \\ s_{\mathrm{tot},2}(-1) &= (w_E - w_I)(u_R + 1) + w_I U + E_c = s_{\mathrm{sat}}. \end{aligned} \tag{D.6}$$

Similar conditions hold for the right-most state.

We can use these conditions to obtain a lower bound on E_c. If we take equation D.2 for $u = u_R$ and add it to equation D.5, then

$$(w_E + w_I)U - 2(w_E - w_I) + 2E_c \leq 2s_{\mathrm{th}},$$

which can be solved for E_c, using equation D.4, and results in equation 7.5.

An upper bound on E_c can be obtained in a similar way by taking equation D.3 for $u = u_L$ and adding it to equation D.6 to yield

$$(w_E + w_I)U + 2(w_E - w_I) + 2E_c \geq 2s_{\mathrm{sat}},$$

from which follows equation 7.6.

Appendix E: Monotonic Line Attractors

E.1 Numerical Tuning of the Weight Matrix. The construction of monotonic line attractors in the literature has usually been based on numerical tuning of the weight matrix (Seung, 1996; Eliasmith, 2005). Here we review the construction of these line attractors and show their relation to the basic monotonic line attractor from section 5.

Let us assume that the weight matrix of the network has rank one, that is, that it can be written as the outer product of two vectors,

$$w(x, x') = a(x)b(x').$$

In this case, all trajectories of the network will flow onto a single dimension. To see that, let us define the variable

$$u(t) = \int_{-1}^{1} dx\, b(x)s(x, t). \tag{E.1}$$

With this definition, the dynamics of the network follow from equation 2.1:

$$\frac{\partial}{\partial t}s(x, t) = -s(x, t) + f(a(x)u(t) + E(x)).$$

In other words, the firing rates $s(x, t)$ of all the neurons are driven by a single time-varying variable $u(t)$, and the dynamics of this network can be investigated by looking at the dynamics of this single variable. The time derivative of equation E.1 yields

$$\frac{du(t)}{dt} = -u(t) + \int_{-1}^{1} dx\, b(x) f(a(x)u(t) + E(x))$$

with steady state

$$u = \int_{-1}^{1} dx\, b(x) f(ua(x) + E(x)). \tag{E.2}$$

The choice of a rank-one weight matrix therefore reduces the task of designing a line attractor to finding a set of parameters $a(x)$, $b(x)$, and $E(x)$ for which this equation holds. Note that for given (smooth) $a(x)$ and $E(x)$, this equation is known as an inhomogeneous Fredholm integral equation of the first kind with the kernel $K(m, x) = f(ma(x) + E(x))$ and the unknown function $b(x)$ (Byron & Fuller, 1969; Press et al., 1992).

The equation can be solved numerically for $b(x)$ if the kernel $K(m, x)$ is invertible (Press et al., 1992). In previous studies, $a(x)$ and $E(x)$ have usually been chosen randomly, ensuring invertibility of the above equation, so that $b(x)$ could be determined numerically. Note that this approach does not require saturating synapses; in fact, the function $f(\cdot)$ could in principle take almost any form.

E.2 Relation to Basic Monotonic Line Attractor and Heterogeneity. The basic monotonic line attractor from section 5 is an analytical solution to equation E.2 and can therefore be regarded as a special case of the rank-one line attractors. However, the analogy to the numerically tuned networks can be taken one step further when considering large networks with heterogeneous neurons. In essence, if the network is large enough, any heterogeneity will average out, so that all rank-one line attractors become equivalent to the basic monotonic line attractor.

Let us assume that every neuron has a different input-output function $s = f_x(s_{\text{tot}})$, marked by its position x. We still assume that the synaptic output eventually saturates; however, we now allow saturation at values different from one. In particular, we are interested in the parameterization

$$s = f_x(s_{\text{tot}}) = \eta(x)\, g(\xi(x)s_{\text{tot}}). \tag{E.3}$$

We assume that the parameters $\eta(x)$ and $\xi(x)$ of this input-output function are randomly drawn at each position x from appropriate probability distributions. Without loss of generality, we assume that both distributions are centered around one. We denote the average input-output function as $\bar{f}(\cdot)$.

We can now construct a basic monotonic line attractor network with weight matrix $w(x, x') = w_E$ and input-output function $\bar{f}(\cdot)$ as described in section 5. For this network, the synaptic inputs in the continuous attractor state are given by

$$\begin{aligned} s_{\text{tot}}(x) &= \int dx' w_E s(x') + E(x) \\ &= \int dx' w_E \bar{f}(s_{\text{tot}}(x')) + E(x). \end{aligned}$$

Since the synaptic input $s_{\text{tot}}(x')$ is statistically independent from the randomly drawn input-output functions $f_{x'}(\cdot)$, the integral remains the same if we replace the average input-output function by the heterogeneous input-output function,

$$s_{\text{tot}}(x) = \int dx' w_E f_{x'}(s_{\text{tot}}(x')) + E(x).$$

Accordingly, the total synaptic input $s_{\text{tot}}(x)$ to each neuron in a heterogeneous network with input-output functions $s = f_x(s_{\text{tot}})$ is the same as in the basic monotonic line attractor network. Since the above equations are steady-state equations for the synaptic inputs, both the heterogeneous and homogeneous networks must be in a fixed point. Hence, a network with heterogeneous neurons designed along the lines above also has a line attractor.

The resulting heterogeneous network is formally equivalent to a network of homogeneous neurons with a weight matrix $w(x, x') = w_E \eta(x)\xi(x')$, which can be seen by plugging equation E.3 into equation 3.1.[3] The reason, of course, is that in large-scale networks, this random heterogeneity is simply smoothed out, creating a dynamics that is equivalent to the dynamics of the basic monotonic line attractor network. Hence, any network with a rank-one matrix built along these lines can be regarded as equivalent to the basic monotonic line attractor.

For discrete networks, the heterogeneity will not be averaged out exactly since we sum only over a finite number of neurons. In this case, there is no guarantee to obtain N fixed points, and the dynamics of the homogeneous network are only an approximation to those of the heterogeneous network. The resulting mismatches will then lead to drifts of the memory states

[3] A similar argument can be made for arbitrary Toeplitz-type networks. Accordingly, line attractors can potentially be built for any weight matrix that can be written as $w(x, x') = \eta(x)k(x - x')\xi(x')$ if $\eta(x)$ and $\xi(x')$ are random numbers and $k(x - x')$ is sufficiently smooth.

between stable fixed points, as has been noted before (Seung, 1996). In summary, the numerical tuning of monotonic line attractor networks allows a large variety of networks, some of which are mathematically identical to the basic monotonic line attractor network in section 5.

Acknowledgments

We thank Ranulfo Romo for a fruitful and inspiring collaboration. This work was supported by a Swartz Foundation Fellowship to C.K.M. and NIH grant R01MH067991 to C.D.B.

References

Aksay, E., Baker, R., Seung, H. S., & Tank, D. W. (2003). Correlated discharge among cell pairs within the oculomotor horizontal velocity-to-position integrator. *J. Neurosci.*, *23*, 10852–10858.

Aksay, E., Gemkrelidze, G., Seung, H. S., Baker, R., & Tank, D. W. (2001). In vivo intracellular recording and perturbation of persistent activity in a neural integrator. *Nat. Neurosci.*, *4*, 184–193.

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybern.*, *27*(2), 77–87.

Ben-Yishai, R., Bar-Or, R. L., & Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. *Proc. Natl. Acad. Sci. USA*, *92*, 3844–3848.

Brody, C. D., Hernandez, A., Zainos, A., & Romo, R. (2003). Timing and neural encoding of somatosensory parametric working memory in macaque prefrontal cortex. *Cereb. Cortex*, *13*, 1196–1207.

Byron, F. W., & Fuller, R. W. (1969). *Mathematics of classical and quantum physics*. Reading, MA: Addison-Wesley.

Cannon, S. C., Robinson, D. A., & Shamma, S. (1983). A proposed neural network for the integrator of the oculomotor system. *Biol. Cybern.*, *49*, 127–136.

Compte, A., Brunel, N., Goldman-Rakic, P. S., & Wang, X.-J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cereb. Cortex*, *10*, 910–923.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

Droulez, J., & Berthoz, A. (1991). A neural network model of senoritopic maps with predictive short-term memory properties. *Proc. Natl. Acad. Sci. USA*, *88*, 9653–9657.

Durstewitz, D. (2003). Self-organizing neural integrator predicts interval times through climbing activity. *J. Neurosci.*, *23*, 5342–5353.

Eliasmith, C. (2005). A unified approach to building and controlling spiking attractor networks. *Neural Comput.*, *17*(6), 1276–1314.

Ermentrout, B. (1994). Reduction of conductance based models with slow synapses to neural nets. *Neural Comp.*, *6*, 679–695.

Fransen, E., Tahvildari, B., Egorov, A. V., Hasselmo, M. E., & Alonso, A. A. (2006). Mechanism of graded persistent cellular activity of entorhinal cortex layer V neurons. *Neuron*, *49*, 735–746.

Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge: Cambridge University Press.

Koulakov, A. A., Raghavachari, S., Kepecs, A., & Lisman, J. E. (2002). Model for a robust neural integrator. *Nat. Neurosci.*, *5*(8), 775–782.

Loewenstein, Y., & Sompolinsky, H. (2003). Temporal integration by calcium dynamics in a model neuron. *Nat. Neurosci.*, *6*, 961–967.

Machens, C. K., Romo, R., & Brody, C. D. (2005). Flexible control of mutual inhibition: A neural model of two-interval discrimination. *Science*, *307*, 1121–1124.

Major, G., & Tank, D. (2004). Persistent neural activity: Prevalence and mechanisms. *Curr. Opin. Neurobiol.*, *14*, 675–684.

Miller, P., Brody, C. D., Romo, R., & Wang, X.-J. (2003). A recurrent network model of somatosensory parametric working memory in the prefrontal cortex. *Cereb. Cortex*, *13*(11), 1208–1218.

Morishita, I., & Yajima, A. (1972). Analysis and simulation of networks of mutually inhibiting neurons. *Kybernetik*, *11*, 154–165.

Nakamura, K. (1999). Auditory spatial discriminatory and mnemonic neurons in rat posterior parietal cortex. *J. Neurophysiol.*, *82*(5), 2503–2517.

O'Keefe, J. (1979). A review of the hippocampal place cells. *Prog. Neurobiol.*, *13*, 419–439.

Pinto, D. J., & Ermentrout, G. B. (2001). Spatially structured activity in synaptically coupled neuronal networks: II. Lateral inhibition and standing pulses. *SIAM J. Appl. Math.*, *62*, 226–243.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes in C*. Cambridge: Cambridge University Press.

Quirk, G. J., Muller, R. U., & Kubie, J. L. (1990). The firing of hippocampal place cells in the dark depends on the rat's recent experience. *J. Neurosci.*, *10*, 2008–2017.

Rainer, G., & Miller, E. K. (2002). Timecourse of object-related neural activity in the primate prefrontal cortex during a short-term memory task. *Eur. J. Neurosci.*, *15*, 1244–1254.

Renart, A., Brunel, N., & Wang, X.-J. (2003). Mean-field theory of recurrent cortical networks: From irregularly spiking neurons to working memory. In J. Feng (Ed.), *Computational neuroscience: A comprehensive approach*. Boca Raton, FL: CRC Press.

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

Samsonovich, A., & McNaughton, B. L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *J. Neurosci.*, *17*, 5900–5920.

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

Seung, H. S., Lee, D. D., Reis, B. Y., & Tank, D. W. (2000). Stability of the memory of eye position in a recurrent network of conductance-based model neurons. *Neuron*, *26*, 259–271.

Shriki, O., Hansel, D., & Sompolinsky, H. (2003). Rate models for conductance-based cortical neural networks. *Neural Comp.*, *15*, 1809–1841.

Singh, R., & Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *J. Neurosci.*, *26*, 3667–3678.

Skaggs, W., Knierim, J. J., Kudrimoti, H. S., & McNaughton, B. L. (1995). A model of the neural basis of the rat's sense of direction. In G. Tesauro,

D. S. Touretzky & T. K. Leen (Eds.), *Advances in neural information processing systems*, *7* (pp. 173–180). Cambridge, MA: MIT Press.

Smyrnis, N., Taira, M., Ashe, J., & Georgopoulos, A. P. (1992). Motor cortical activity in a memorized delay task. *Exp. Brain Res.*, *92*, 139–151.

Taube, J. S., & Bassett, J. P. (2003). Persistent neural activity in head direction cells. *Cereb. Cortex*, *13*, 1162–1172.

Tsodyks, M. (1999). Attractor neural network models of spatial maps in hippocampus. *Hippocampus*, *9*, 481–489.

Wang, X.-J. (1999). Synaptic basis of cortical persistent activity: The importance of NMDA receptors to working memory. *J. Neurosci.*, *19*, 9587–9603.

Wang, X.-J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends in Neurosciences*, *24*, 455–463.

Wilson, H. R., & Cowan, J. D. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, *13*, 55–80.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble. *J. Neurosci.*, *16*(6), 2112–1226.

Zipser, D., Brandt, K., Littlewort, G., & Fuster, J. (1993). A spiking network model of short-term active memory. *J. Neurosci.*, *13*, 3406–3420.

Received July 31, 2006; accepted January 28, 2007.