

Does the Entorhinal Cortex use the Fourier Transform?

J. Orchard^{*†}, Hao Yang[†], Xiang Ji^{*†}

January 16, 2013

Abstract

In 2005, Hafting et al. reported that some neurons in the entorhinal cortex (EC) fire bursts when the animal occupies locations organized in a hexagonal grid pattern in their spatial environment. Previous to that, place cells had been observed, firing bursts only when the animal occupied a particular region of the environment. Both of these types of cells exhibit theta-cycle modulation, firing bursts in the 4-12Hz range. In particular, grid cells fire bursts of action potentials that precess with respect to the theta cycle, a phenomenon dubbed “theta precession”. Since then, various models have been proposed to explain the relationship between grid cells, place cells, and theta precession. However, most models have lacked a fundamental, overarching framework. As a reformulation of the pioneering work of Welday et al. (2011), we propose that the EC is implementing its spatial coding using the Fourier Transform. We show how the Fourier Shift Theorem relates to the phases of velocity-controlled oscillators (VCOs), and propose a model for how various other spatial maps might be implemented (eg. border cells). Our model exhibits the standard EC behaviours: grid cells, place cells, and phase precession, as borne out by theoretical computations and spiking-neuron simulations. We hope that framing this constellation of phenomena in Fourier Theory will accelerate our understanding of how the EC – and perhaps the hippocampus – encodes spatial information.

^{*}Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, Ontario, Canada

[†]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada

1 Introduction

Some neurons in the entorhinal cortex (EC) exhibit spatial firing patterns (Hafting et al., 2005). These neurons, called “grid cells”, spike preferentially when the animal is at points arranged in a hexagonal grid pattern. Before that, neurons in the hippocampus were found to activate when the animal was in a particular location in the environment. These neurons are called “place cells”.

Both types of cells, place cells and grid cells, are modulated by the theta rhythm, a pattern of activity that oscillates at between 4 and 12 Hz. Moreover, the frequency at which grid cells oscillate is also influenced by the animal’s movement. As the animal is moving, the frequencies increase slightly. But running direction is also a factor, and frequency increases also depend on how close the animal’s velocity vector is to a particular grid cell’s preferred direction vector. If the animal moves in the preferred direction, the frequency increases more, whereas in the opposite direction, the frequency increases only marginally. The term “velocity-controlled oscillator”, or VCO, denotes a neuron or population of neurons whose activity oscillates, but at a frequency that is modulated by velocity.

Combining the ideas of VCOs and grid cells, researchers proposed that the grid patterns might arise from an interference pattern between VCOs. As the animal moves, these VCOs take on slightly different frequencies, and hence their relative phases change.

Phase is the time integral of frequency. Moreover, the phase difference, ϕ , between two oscillators is the

integral of their frequency difference,

$$\phi(t) = \int_0^t (\omega_1(\tau) - \omega_2(\tau)) d\tau ,$$

where ω_1 and ω_2 are the frequencies of the two VCOs. If the frequency of the VCOs are linear functions of velocity, then as the animal moves the phase of the VCO can be written

$$\begin{aligned} \phi(t) &= \int_0^t c_1 v(\tau) - c_2 v(\tau) d\tau \\ &= (c_1 - c_2) \int_0^t v(\tau) d\tau \\ &= (c_1 - c_2) x(t) , \end{aligned}$$

where c_1 and c_2 are scalar constants. Hence the phase difference $\phi(t)$ is proportional to total displacement $x(t)$.

By combining (adding) two VCOs with different frequencies, the result is a beat interference pattern that generates periods of constructive and destructive interference as their phase difference evolves (Blair et al., 2008). Since phase and position are linked, this interference pattern overlays the animal’s spatial environment. Combining three VCOs (that differ in preferred direction by multiples of 60°) tends to create a hexagonal grid interference pattern (Burgess et al., 2007).

A satisfactory explanation of the relationship between grid cells and place cells remained unclear. As recently as 2008, researchers had only a handful of ideas of how grid cells might combine to produce place cells (Moser et al., 2008). Some have experimented with combining a random selection of grid cells to produce place-cell like behaviour (Fuhs and Touretzky, 2006; Solstad et al., 2006). Others suggested that a sum of grid cells could create place cells, but offered only vague justification (O’Keefe and Burgess, 2005; McNaughton et al., 2006). A more detailed proposal argued that place cells resulted from the Moiré interference patterns between small-scale grid patterns (Blair et al., 2007). However, their method involves intricate rescaling of so-called “theta cell” grids, which the authors point out as “potentially a serious limitation” (Blair et al.,

2007). The method seems overly complicated considering it comprises a linear sum of periodic functions. In our opinion, Fourier Theory is a better choice than Moiré interference patterns.

A recent study concluded that the distributed encoding of grid cells formed a more efficient representation than the same number of place cells (Mathis and Herz, 2012). That work is interesting, but does not discuss the mechanisms underpinning these various cell types. A comprehensive review of the various proposed models can be found in (Zilli, 2012).

A spiking-neuron based model of path integration used Gaussian surfaces to represent place cells, but encoded these Gaussians by their Fourier coefficients (Conklin and Eliasmith, 2005). This implementation takes advantage of the Fourier Shift Theorem (discussed later), moving the Gaussian pattern of excitation around by applying phase shifts to the Fourier coefficients. However, their model does not address grid cells. Can the Fourier Shift Theorem be used in conjunction with grid cells?

In 2011, Welday et al. proposed a more complete theory of the mechanisms combining grid cells, place cells, and phase precession (Welday et al., 2011). Their model involves a bank of VCOs arranged in a 2-dimensional (2-D) array, where one dimension spans a variety of preferred directions, and the other dimension represents the degree to which frequency is increased by movement. The left pane of Fig. 1 is a recreation of a portion of Fig. 7 from their paper.

In their firing-rate model, each VCO is modelled as a ring oscillator with a wave of activity that cycles around at (or near) theta frequency. Hence, each neuron on the ring activates at a particular phase. According to their paper, connecting a read-out neuron to all the phase-matched neurons of a given row produces a place cell. Similarly, choosing only three phase-matched neurons from a row, but with preferred directions separated by 120° , yields a grid cell. Finally, choosing all the phase-matched neurons from rings with the same preferred direction vector can generate a border cell.

While the ideas presented in that paper have merit, the authors’ explanations for their claims are somewhat disconnected and hard to follow. In this paper, we build on their model, but re-formulate it into a

coherent and elegant framework using Fourier Theory.

2 Fourier Model

The bank-of-oscillators model states that a VCO's frequency depends on two parameters: the speed of the animal, and the cosine of the animal's velocity vector with the VCO's preferred direction. These VCOs can be organized into a 2-D space in which one axis enumerates preferred directions, and the other axis represents the influence of speed (labeled "slope of speed modulation" in Fig. 1A).

Another, perhaps more intuitive way of presenting the same 2-D parameter space is to use polar coordinates, as shown in Fig. 1B. In this view, the direction of displacement from the origin indicates the preferred direction, and the distance from the origin multiplies that effect. In addition, we will think of each VCO as a small vector that oscillates about its centre point. Usually, we will consider these oscillators to be unit-vectors, and focus our attention on their phase.

Consider the VCO located at **A** in Fig. 2A, 4 units from the origin, in the direction of 30° . If the animal moves in that direction, the VCO will experience an increased frequency compared to the VCO at the origin. More precisely, the VCO's frequency will increase by an amount proportional to $\mathbf{v} \cdot \mathbf{A}$, the dot-product of $\mathbf{A}$ and $\mathbf{v}$. After a time t , the difference in phase will be

$$\phi(t) = \int_0^t \mathbf{A} \cdot \mathbf{v}(\tau) d\tau = \mathbf{A} \cdot \int_0^t \mathbf{v}(\tau) d\tau = \mathbf{A} \cdot \mathbf{x}(t).$$

That is, the phase difference will be $\mathbf{A} \cdot \mathbf{x}(t)$. Since ϕ is an angle, it is convenient to depict it as a vector on the unit circle. Then we can write this "phase vector" using Euler's formula,

$$(\cos \phi, \sin \phi) \equiv \cos \phi + i \sin \phi = \exp(i\phi),$$

where $i = \sqrt{-1}$. Thus, we can represent our phase difference as

$$(\cos \phi, \sin \phi) = \exp(i\mathbf{A} \cdot \mathbf{x}).$$

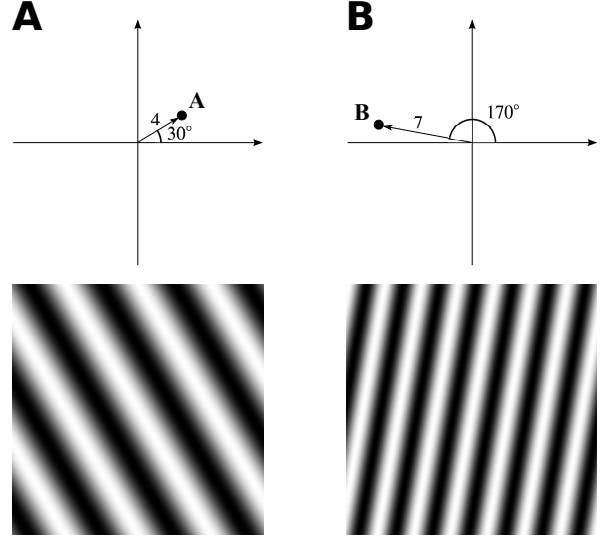


Figure 2: Trigonometric wave fronts. The bottom row shows the real part of the Fourier basis function corresponding to a single non-zero Fourier coefficient set to 1.

As we can see, the components of the phase vector trace out sine and cosine wave fronts that are fixed in the animal's environment. The bottom row of Fig. 2A shows the wave front corresponding to the VCO located at **A**. A different VCO at location **B** traces out a different wave front, as shown in Fig. 2B.

Considering that the animal has many such VCOs, what happens if we combine them all into a sum, as if a single read-out node was receiving the phase vectors from all the VCOs and adding them together? The value of the read-out node would be

$$\mathbf{p}(\mathbf{x}) = \sum_{k,\ell} \exp(i\mathbf{d}_{k,\ell} \cdot \mathbf{x}), \quad (1)$$

where $\mathbf{d}_{k,\ell}$ is the location of a VCO, and the subscripts k and ℓ index distance from the origin and orientation, respectively. An image created using this simple method is shown in Fig. 3. The activity of this read-out node corresponds to a spatial map akin to a place cell. Why is that? The answer has to do with the fact that (1) almost looks like an inverse Discrete

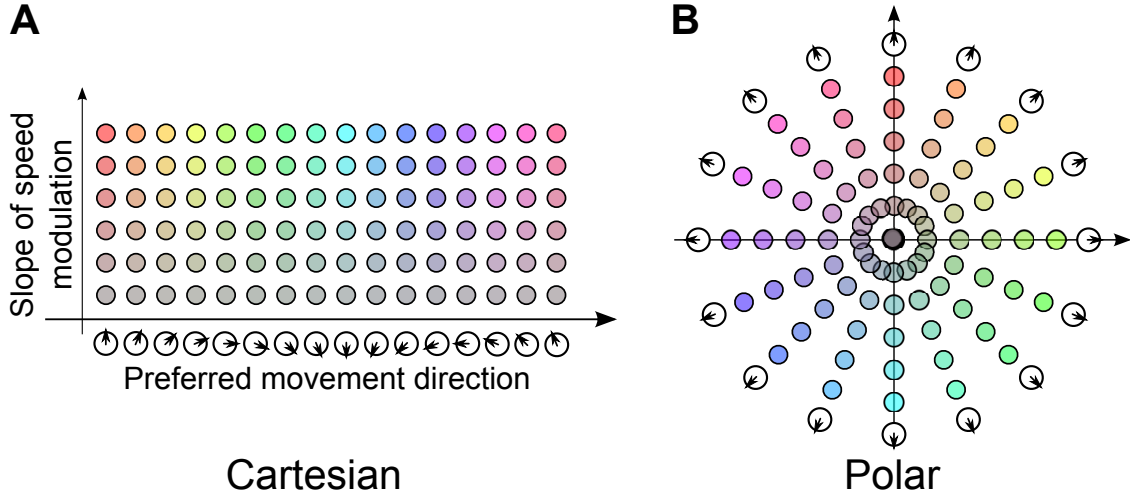


Figure 1: Cartesian versus polar representation of VCOs. The Cartesian arrangement is derived from part of Fig. 7 in (Welday et al., 2011). The polar arrangement consists of a number of “propellers”, lines of VCOs that pass through the origin.



Figure 3: The spatial function traced out by adding together the whole bank of VCOs.

Fourier Transform¹. In the following sections, we review the Fourier Transform and outline the benefits of thinking about the EC in terms of this powerful mathematical tool.

¹It would be the IDFT, except that the summation indices correspond to polar coordinates, whereas the IDFT indices use rectangular spacing.

2.1 Fourier Theory Basics

We will develop our argument using the Discrete Fourier Transform (DFT), but we point out that analogous properties exist for the continuous-domain Fourier Transform (Oppenheim and Schaffer, 1999).

An efficient way to write the Fourier Transform is to use complex numbers. Recall that we have two ways of denoting a complex number: using the Cartesian for $a + ib$, or the polar form $r \exp(i\phi)$. In the Cartesian form, we refer to a as the *real part*, and b as the *imaginary part*. In the polar (or exponential) form, we call r the *modulus*, and ϕ the *phase angle*. This notation carries with it several advantages. The product of two complex numbers has a simple algebraic analog when using the polar representation. In particular, multiplying $r_1 \exp(i\phi_1)$ and $r_2 \exp(i\phi_2)$ is consistent with the algebraic rules for exponential functions, giving $r_1 r_2 \exp(i(\phi_1 + \phi_2))$. In words, you multiply their moduli, and add their phases.

We will start out with a 1-D function. Consider a sampled function f with N samples indexed $n = 0, \dots, N - 1$,

$$f = [f_0, f_1, \dots, f_{N-2}, f_{N-1}] .$$

The DFT of f is

$$F_k = \sum_{n=0}^{N-1} f_n \exp\left(-2\pi i \frac{nk}{N}\right), \quad k = 0, \dots, N-1.$$

Each complex number F_k is called a Fourier coefficient. We can also denote the transform using $F = \text{DFT}(f)$. In essence, the DFT is a frequency decomposition; it takes a spatial signal and represents it as a sum of wave fronts of various frequencies (and orientations, in higher dimensions). Each Fourier coefficient occupies a different location in the frequency domain. Each location in the frequency domain represents a different wave front. The value of a Fourier coefficient, F_k , represents the contribution of its wave front. The coefficient F_0 has a special name; it is called the DC, and it always corresponds to the zero frequency and is located at the origin of the frequency domain. It is important to understand that the Fourier basis functions are N -periodic. That is,

$$\exp\left(-2\pi i \frac{(n+N)k}{N}\right) = \exp\left(-2\pi i \frac{nk}{N}\right).$$

In other words, it is perfectly valid to refer to F_{-1} , since $F_{-1} = F_{N-1}$. Because of this periodicity, we can shift our array of Fourier coefficients and visualize them as being centred on the DC. For example, if N is 33, then we list our Fourier coefficients using $[F_{-16}, \dots, F_{-1}, F_0, F_1, \dots, F_{16}]$. Likewise, use of the DFT implicitly assumes that f is also periodic and can be indexed in the same manner. Thus, an equivalent formulation of the DFT is

$$F_k = \sum_{n=-\tilde{N}}^{\tilde{N}} f_n \exp\left(-2\pi i \frac{nk}{N}\right), \quad k = -\tilde{N}, \dots, \tilde{N},$$

where we assume for simplicity that N is odd² and use the symbol $\tilde{N}$ to represent $\lfloor \frac{N}{2} \rfloor$, where the delimiters $\lfloor \cdot \rfloor$ denote rounding toward zero. We will use this equivalent, centred version of the DFT throughout this paper.

²We could remove the assumption that N is odd, and sum over $n = -\tilde{N}, \dots, \tilde{N} - 1$.

The Fourier Shift Theorem tells us how shifting (translating) a signal influences its Fourier coefficients. Suppose that F_k are the Fourier coefficients of a signal f_n . Consider a shifted version, f_{n-d} , and its Fourier coefficients, G_k . What is the relationship between G_k and F_k ? The Fourier transform of f_{n-d} can be written

$$G_k = \sum_{n=-\tilde{N}}^{\tilde{N}} f_{n-d} \exp\left(-2\pi i \frac{nk}{N}\right).$$

Using the change of variables $m = n - d$, we get

$$G_k = \sum_{m=-\tilde{N}-d}^{\tilde{N}-d} f_m \exp\left(-2\pi i \frac{(m+d)k}{N}\right).$$

Both $\exp(\cdot)$ and f are periodic (by assumption, for f), so we can shift the summation range without changing its value. Moreover, the exponential function can be split into two components, one containing m , while the other can be pulled out of the summation,

$$\begin{aligned} G_k &= \exp\left(-2\pi i \frac{dk}{N}\right) \sum_{m=-\tilde{N}}^{\tilde{N}} f_m \exp\left(-2\pi i \frac{mk}{N}\right) \\ G_k &= \exp\left(-2\pi i \frac{dk}{N}\right) F_k, \quad k = -\tilde{N}, \dots, \tilde{N}. \end{aligned}$$

This is the Fourier Shift Theorem, and it tells us that the Fourier coefficients of the shifted signal, f_{n-d} , can be derived from the coefficients of the original signal by simply multiplying each coefficient by a phase-shift, where the amount of the phase-shift is a linear function of the frequency index k . We will unpack this theorem more as we progress.

As an example relevant to our purposes, suppose we have a rat that moves along a small corridor, and that we have broken the corridor into N blocks, indexed $n = -\tilde{N}, \dots, \tilde{N}$. We can represent the location of the rat using an array, δ , indicating where the rat is by placing a 1 in the element corresponding to the rat's location, and setting all the other elements to zero. As the rat moves along the corridor, the array δ changes so that the location of the 1 reflects which block the rat occupies. By this definition, δ is the

Kronecker Delta,

$$\delta_{mn} = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases}$$

where the subscript m represents the index of the block containing the rat, and the subscript n indexes the elements of the array δ_m .

The DFT of δ has special properties. When the rat is a position m , the DFT of δ_m is,

$$\begin{aligned} G_k &= \sum_{n=-\tilde{N}}^{\tilde{N}} \delta_{mn} \exp\left(-2\pi i \frac{nk}{N}\right) \\ &= \exp\left(-2\pi i \frac{mk}{N}\right). \end{aligned}$$

These Fourier coefficients all have a modulus of 1, and their phases vary linearly with k (the frequency index). That is, if $G_k = r_k \exp(i\phi_k)$, then $r_k = 1$ and

$$\phi_k = -2\pi \frac{mk}{N}.$$

The modulus and phase of δ_2 are shown in Fig. 4. Notice that the phases in Fig. 4 form a line (or ramp) as you move along the frequency axis, although the phases are wrapped into the range $[-\pi, \pi]$. The slope of the line is $(-2\pi m/N)$, so the larger m is, the steeper the slope. The sign of the slope reflects whether m is positive or negative (which direction the rat moved along the corridor).

This property of the DFT is similar to a more general concept known as the Fourier Shift Theorem, in which a function can be shifted by simply multiplying its Fourier coefficients by a linear phase function (or “phase ramp”) like that depicted in Fig. 4. If we start with the Fourier coefficients of δ_0 , denoted F_k , then we can get the Fourier coefficients of δ_m , denoted G_k , by multiplying F_k by the phase ramp $\exp(-2\pi imk/N)$. That is, $G_k = \exp(-2\pi imk/N) F_k$. Applying such a phase ramp can shift any function, not just our δ functions.

The DFT is an invertible transform, and the inverse DFT (IDFT) yields back the original function. For example, $\text{IDFT}(\text{DFT}(\delta_m)) = \delta_m$. With this in mind, we can represent any shifted version of δ_0 by

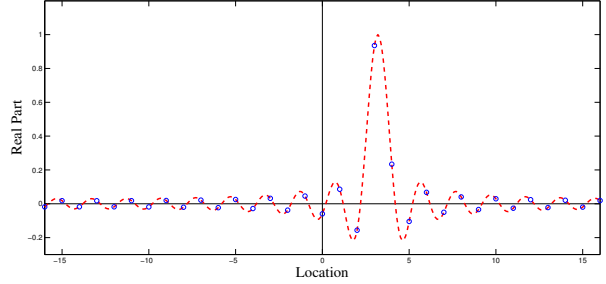


Figure 5: Fourier interpolation of δ_0 shifted to the right by 3.2 units. The dotted red line shows the continuous-domain reconstruction from the phase-shifted Fourier coefficients. The blue circles show the samples produced using the IDFT.

multiplying its Fourier coefficients by a phase ramp of the desired slope. In other words, the slope of the phase ramp in the Fourier coefficients indicates the shift applied to δ_0 . In this way, the phases of the Fourier coefficients encode the position of the rat.

What if the slope $(-2\pi m/N)$ of the phase ramp along k corresponds to a value of m that is not an integer? We cannot expect the IDFT to give us δ_m exactly, since that is only defined for integer values of m . As it turns out, non-integer values of m yield δ -like signals. The operation is called “Fourier interpolation”, and corresponds to summing the continuous-domain wave fronts and sampling the resulting function. Figure 5 illustrates a non-integer shift, and how the samples computed by the IDFT relate to the underlying continuous-domain Fourier reconstruction.

All of this theory extends trivially to 2-D domains (and higher). A Kronecker delta (or any function, for that matter) on a 2-D domain can be shifted by multiplying its Fourier coefficients by a phase ramp in 2-D.

2.2 Entorhinal Cortex

Here we outline our model of how the Fourier Transform relates to the navigational function of the entorhinal cortex (EC).

We propose that each VCO in the polar arrangement corresponds to a Fourier coefficient in a 2-D

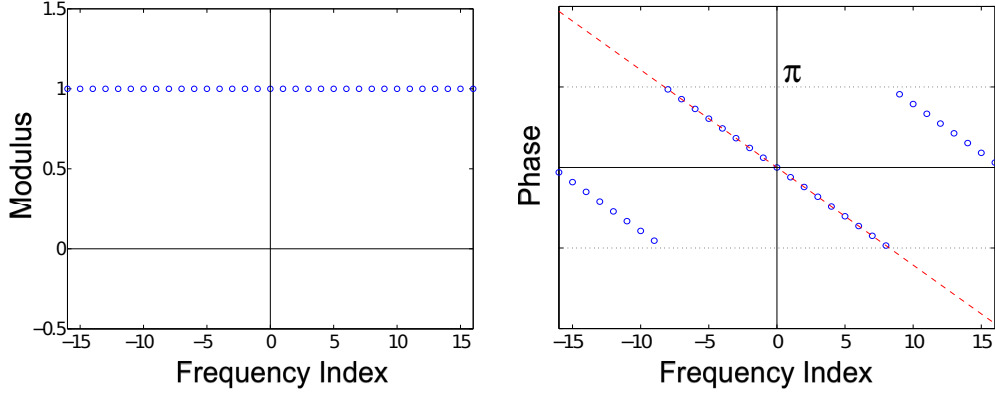


Figure 4: Fourier Transform of δ_2 . The moduli (left) of the 33 Fourier coefficients are all 1. The phase (right) is linear; the dashed red line shows a linear function, while the blue dots show the same linear function wrapped into the range $[-\pi, \pi]$.

frequency domain, and is used to modulate its corresponding wave front to build spatial maps like grid cells and place cells.

Why? As the animal moves around, the evolving phase differences between the DC and the other VCOs encode the animal’s location in a manner convenient for applying the same positional shift to any spatial map using the Fourier Shift Theorem. The key idea is that if spatial maps of the hippocampus and EC are encoded using a Fourier representation, then the activity at the animal’s location in each map can be extracted by rotating the phase of each Fourier coefficient by the corresponding VCO phase.

But where are the Fourier coefficients that make up these spatial maps? They manifest in the connections to the VCOs.

Let’s be more concrete. Figure 6 shows a number of spatial maps similar to those elicited by place cells, grid cells, and “border” cells. Consider two neurons, one is a place cell with its spatial activity map shown in A, and the other is a grid cell, its spatial activity map shown in B. As the animal moves throughout its environment, the place cell and grid cell change their activity according to where the animal is in those two maps. Each of those spatial maps has a very simple Fourier transform, shown above it in the figure. The shaded Fourier coefficients all have a value of 1, while

all the others are zero (Fig. 6D is an exception to that, discussed later). However, the shaded nodes in the figure actually represent connections to VCOs. That is, the connections to the VCOs are the Fourier coefficients of the spatial map.

In the case of our place cell, all of its Fourier coefficients are 1, so it is connected to all the VCOs and simply adds all their phase vectors together. This is the same operation described in equation (1). For the grid cell, only three of its Fourier coefficients are 1, so it adds together the phase vectors from those three VCOs. Again, the *connections* to the bank of VCOs dictate the contents of the spatial map.

Movement throughout those maps is represented by the phase state of the VCOs. As we already discussed, animal motion causes the VCO frequencies to change in a way that induces a phase ramp. Those ramped phase vectors get multiplied by the connection strengths on their way to the place cell (or grid cell). Hence, the place cell (or grid cell) in essence adds up a set of Fourier components each modulated by a phase ramp. The place cell and grid cell have fixed connections to the VCOs, yet their activity changes as the animal moves because the phase state of the VCO bank shifts the spatial map using the Fourier Shift Theorem.

The “border” cell shown in Fig. 6D follows the same

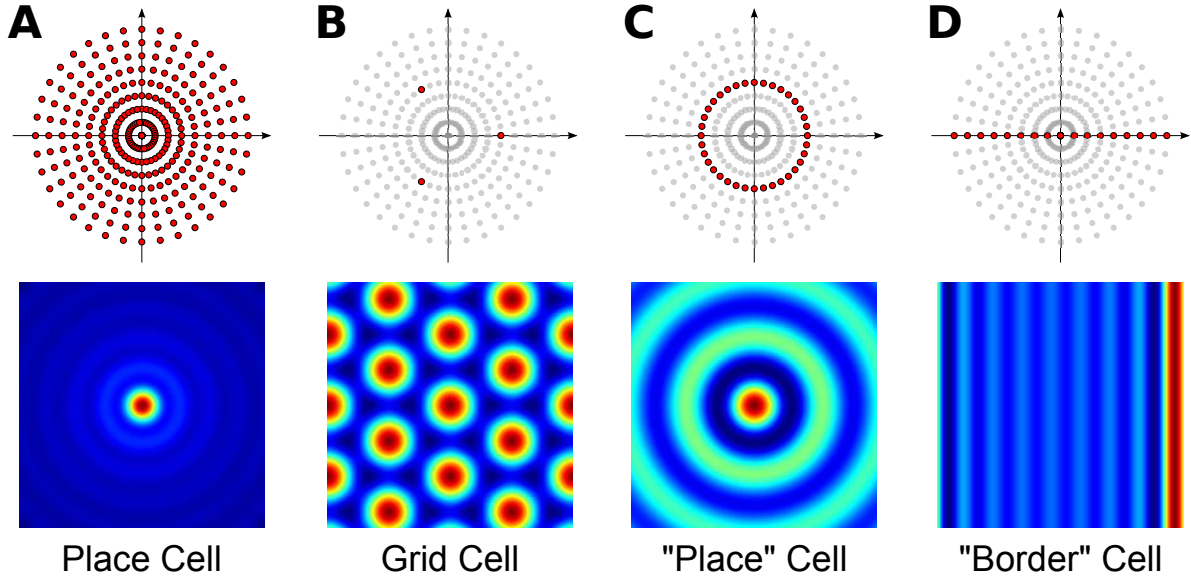


Figure 6: Sample spatial maps (bottom row) and the VCOs used to generate them (top row).

rationale, but its Fourier coefficients are not all 1s. If the shaded coefficients were all 1s, then the vertical line of activity would be centred in the spatial map. For the sake of comparison to (Welday et al., 2011), we shifted the spatial map to the edge of the field. We did so by including a phase ramp in the Fourier coefficients. In other words, the connections from the VCOs to the “border” cell induced an additional phase shift. The phase ramp of the connections relate to the location of the bar in the spatial map, while the phase ramp in the VCOs relate to the animal’s location.

For the sake of comparison, we included Figure 6C to show the “place” cell that was proposed in (Welday et al., 2011), using only the VCOs from a given spatial frequency (but all orientations).

Again, each VCO can be thought of as a phase-shift coefficient for a 2-D frequency phase ramp. The animal maintains a number of spatial maps of their environment, such as place fields, grid fields, and even more elaborate fields like boundary fields, etc. The actual form of the spatial map is determined by the connection weights that connect the VCO nodes to the map cells. Each map cell (eg. place cell, grid cell,

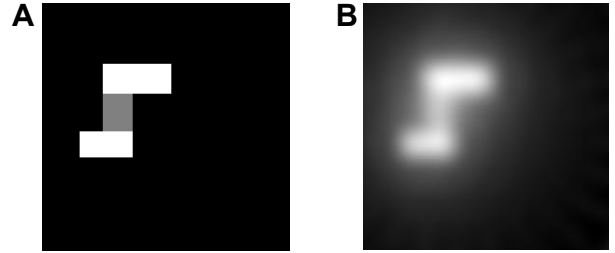


Figure 7: Example of a general spatial map. The Fourier coefficients of the ideal spatial map in A where used to set connection weights from a bank of VCOs (18 propellers, 9 rings) to a readout node. The spatial map of the virtual readout node is shown in B.

etc.) can have a different pattern of activity. As an example, consider the spatial map (image) shown in Fig. 7. The ideal map shown in A was encoded in the connection weights to a bank of VCOs consisting of 18 propellers and 9 rings, like those shown in the top row of Fig. (6). The image in B is a reconstruction of that neuron’s activity as a function of location in the animal’s environment.

Interestingly, grid cells might be a byproduct of the computation of place cells. All the VCOs in Fig. 6a are added together to get the place cell. However, that sum could be done in pieces, one triad of VCOs at a time. As an intermediate step to adding all the VCOs together, all possible triads could be added together to form a bank of grid cells. Then, those grid cells could be added together to form the place cell. We know this two-part process is equivalent to adding the VCOs together all at once because all the processes are linear operations. In fact, the Fourier Transform is a linear operation.

3 Material and Methods

The results shown thus far are from an idealized implementation of the EC and its bank of VCOs. However, we also implemented a partial version of the EC Fourier model using spiking leaky integrate-and-fire (LIF) neurons (Koch, 1999). Here we describe our implementation of the model, outline the challenges, and display results from simulation experiments.

3.1 Neural Engineering Framework

To build our neural network, we used the Neural Engineering Framework (NEF) (Eliasmith and Anderson, 2003), a powerful and versatile platform that has proven useful for large-scale cognitive modeling (Eliasmith et al., 2013). In this framework, data is stored using population coding. A node is a population of LIF neurons with varying parameters so that their tuning curves span a wide range of possibilities. The population of N neurons can encode a value $\mathbf{x}$ in its neural activities using

$$a_n(t) = G_n(\mathbf{x}(t) \cdot \mathbf{e}_n \alpha_n + \beta_n) \quad (2)$$

where $\mathbf{e}_n$ is the encoding vector (preferred direction vector), and α_n and β_n are scalar gain and bias terms that account for the neural climate of the neuron. The input to the function $G_n(\cdot)$ can be thought of as the input current driving neuron n . The function G_n translates the input current to neural activity, either in the form of a firing rate or a series of spikes. In the

case of firing rate, G_n is the steady-state LIF tuning curve,

$$G_n(J) = \begin{cases} \frac{1}{\tau_{\text{ref}} - \tau_m \ln\left(1 - \frac{J_{\text{th}}}{J}\right)} & \text{for } J > J_{\text{th}} \\ 0 & \text{otherwise} \end{cases}$$

where τ_{ref} is the refractory period, τ_m is the membrane time constant, and J_{th} is the threshold current, below which the neuron has a firing rate of zero. On the other hand, if using spikes, then the output of G_n is represented as a sum of time-shifted Dirac delta functions (Oppenheim and Schaffer, 1999),

$$G_n(J_n(t)) = \sum_p \delta(t - t_{np}) ,$$

where t_{np} is the time of the p th spike from neuron n .

When a spike arrives at a synapse, we model the dynamic process of the induced post-synaptic current using an exponential decay. We convolve the incoming spike trains with the post-synaptic filter, $h(t)$,

$$h(t) = \frac{1}{\tau_s} \exp\left(\frac{-t}{\tau_s}\right) \quad (3)$$

and take a weighted sum over all the incoming connections so that the input current arriving at neuron m is

$$J_m(t) = h(t) \star \left[\sum_n w_{nm} \sum_p \delta(t - t_{np}) \right] ,$$

where $\star$ represents convolution, and w_{nm} is the connection weight from neuron n to neuron m .

If we wish to decode the neural activities of a population of neurons, we can compute the optimal linear decoders. We do this by collecting a sampling of inputs, X , and corresponding neural activities, A . That is, each row of X stores a sample input, and each row of A stores the corresponding neural activities for all N neurons (usually stored as firing rates). To decode from our population, we seek the linear weights D that solves

$$\min_D \|AD - X\|_2^2 .$$

Thus, the weights in D perform a linear transformation from the space of neural activities to the space of

input values. This is a linear least-squares problem, and there are multiple ways to compute D . Once we have D , we can decode neural activities to get an estimate of the value being encoded. Moreover, we can decode arbitrary functions of our encoded data by finding the decoders that solve

$$\min_D \|AD - f(X)\|_2^2$$

where $f(X)$ is a function of the encoded values.

With the encoders and decoders, we can transform data from one population P , to another population Q . To do this, we essentially decode the desired function from P and re-encode the result into Q . Collapsing those processes together gives the $N \times M$ weight matrix

$$W = D_P E_Q \alpha_Q$$

where D_P is the matrix that decodes the neural activities from the N neurons in P , and $E_Q \alpha_Q$ stores the scaled preferred direction vectors for all M neurons in Q .

3.2 Oscillators

A recurrently-connected population of neurons can implement a dynamic model of the form

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x})$$

by choosing the recurrent connection weights so that they decode and feed back $\tau_s f(\mathbf{x}) + \mathbf{x}$ (see Eliasmith and Anderson, 2003).

A pertinent example for our purposes is the simple harmonic oscillator,

$$\begin{aligned} \frac{dx}{dt} &= cy \\ \frac{dy}{dt} &= -cx \end{aligned}$$

where c is a scalar. Solutions to this dynamical system include all circular orbits around the origin in the (x, y) -plane. The frequency of oscillation is proportional to c . To implement this behaviour in neurons, we compute the decoders D that decode

$$f(x, y) = [x + \tau_s cy, y - \tau_s cx] .$$

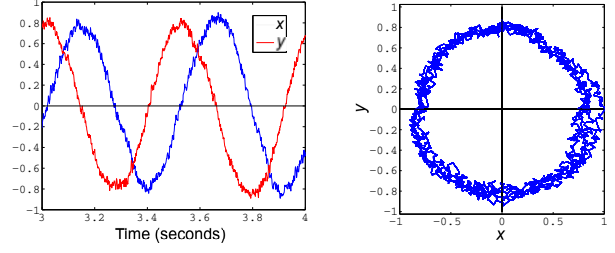


Figure 8: Decoded state of a simple harmonic oscillator implemented using 200 spiking LIF neurons. The plot on the left shows the x and y components over 1 second of time, while the graph on the right plots the phase portrait of x vs. y over the same time frame.

The decoded state is immediately fed back into the population, leading to the trajectory shown in Fig. 8.

A VCO can be constructed by making the population encode a 3-D vector of the form (x, y, θ) , where the x and y components oscillate at frequency (in radians per second) specified by θ . In this case, the decoder would be designed to decode

$$f(x, y, \theta) = [x + \tau_s \theta y, y - \tau_s \theta x]$$

We used a stabilized version of the simple harmonic oscillator by incorporating a unit-vector constraint into the decoder,

$$f(x, y, \theta) = \frac{[x + \tau_s \theta y, y - \tau_s \theta x]}{\|[x + \tau_s \theta y, y - \tau_s \theta x]\|_2} .$$

In our model, the VCOs were modeled using a populations of 300 LIF neurons. We constructed arrays of 17 VCOs to form the propellers seen in the polar arrangement in Fig. 6. As the arrangement dictates, the degree to which the animal's velocity vector influences the frequency of the oscillators depends on where the VCO sits in the plane. Given a velocity vector of $\mathbf{v} \in [-1, 1]^2$, the frequency of VCO at location $\mathbf{d}_n$ is

$$\theta_n = 8 + 1.6 \|\mathbf{v}\|_2 - 1.272 \mathbf{d}_n \cdot \mathbf{v} \quad (4)$$

where the distance from the origin, $\|\mathbf{d}_n\|$, ranges from -1 at one end of a propeller to 1 at the other end. Figure 9 shows how the frequencies vary across a spoke,

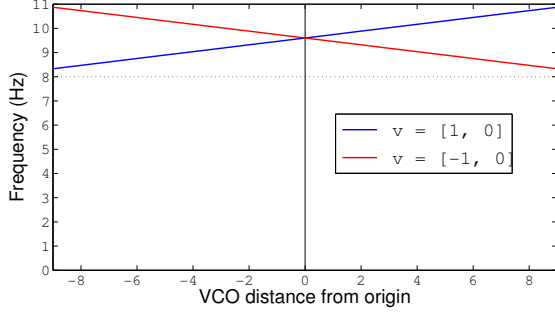


Figure 9: Frequency modulation for VCOs with preferred direction $\mathbf{s}_n = [1, 0]$ and $\mathbf{v}$ either $[1, 0]$ or $[-1, 0]$. The dotted line shows the baseline theta-frequency of 8 Hz. Notice that all VCOs have a frequency above the baseline 8 Hz, no matter which way the rat is running.

and that the frequencies are always above the baseline theta-rhythm of 8 Hz no matter which direction the rat is moving.

3.3 Phase Coupling

The stochastic nature of spiking neurons causes imperfect behaviour of the oscillators. If set to the same frequency and started in phase, perfect oscillators will remain in phase. However, slight errors in frequencies will cause them to drift out of phase as time progresses. This random dephasing can disrupt the phases of the oscillators to the point that the linear trend along k is overwhelmed by noise.

This issue shows up a number of times in the design of this network. For brevity, we outline our solution in detail for one such phase coupling, and include less detail for others.

The absolute phase of the oscillators does not matter. What matters is the phase differences between VCOs. In particular, the phases should follow a linear phase ramp across each propellor. The phases can drift, as long as they maintain their relative phases, maintaining a linear trend across the array.

One way to stabilize the relative phases is to couple the oscillators to each other. We reasoned that the system should be free to allow any linear slope in

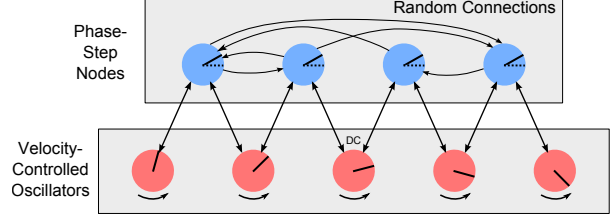


Figure 10: Array of velocity-controlled oscillators. Each adjacent pair of VCOs is coupled by a phase-step node. The phase-step nodes are randomly connected to each other to arrive at a consensus for what the phase difference should be between adjacent VCOs.

phase, but discourage other phase differences.

We introduced an array of neural populations to couple the oscillators. We call these nodes “phase step” nodes. Figure 10 shows the array of oscillators, and their connections to an array of phase-step coupling nodes. Each adjacent pair of oscillators is connected to the same phase-step node. Each phase-step node contains 500 LIF neurons, and represents a 6-dimensional vector of the form $(a, b, \alpha, \beta, c, s)$, where (a, b) and (α, β) are the states of the two connecting oscillators, and (c, s) represents a phase difference of ϕ , where $c = \cos \phi$ and $s = \sin \phi$.

Since the phase should change linearly, each pair of adjacent oscillators should have the same phase difference, or phase step. The phase-step array creates a consensus for this phase difference. Each coupling node decodes the phase difference, $(\tilde{c}, \tilde{s})$, between its afferent VCOs using

$$(\tilde{c}, \tilde{s}) = (a, b) \cdot \overline{(\alpha, \beta)},$$

where (a, b) and (α, β) are the oscillator states, and $\overline{(\alpha, \beta)}$ is the conjugate of (α, β) . If $a + ib = \exp(i\phi_k)$ and $\alpha + i\beta = \exp(i\phi_{k+1})$, then

$$\begin{aligned} \tilde{c} + i\tilde{s} &= \exp(i\phi_k) \exp(-i\phi_{k+1}) \\ &= \exp(i(\phi_k - \phi_{k+1})) \end{aligned}$$

represents the phase difference between the oscillators.

Note, however, that the computation is done by the network entirely in Cartesian coordinates, using

$$\begin{aligned}\tilde{c} + i\tilde{s} &= (a + ib) \cdot (\alpha - i\beta) \\ &= (a\alpha + b\beta) + i(-a\beta + b\alpha).\end{aligned}$$

Each phase-step node decodes $(\tilde{c}, \tilde{s})$ and projects it out to a random subset of other phase-step nodes (including itself). Thus, each phase-step node receives $(\tilde{c}, \tilde{s})$ from a number of other nodes, each equally weighted with all the weights adding to 1. This weighted-average consensus gets stored in the (c, s) -components of each phase-step node's state.

Recall that our phase-step nodes store vectors of the form $(a, b, \alpha, \beta, c, s)$. In a perfect world, the VCO states (a, b) and (α, β) would differ in phase by exactly (c, s) . However, there is always some error. To reduce the error, each phase-step node projects phase adjustments back to their connected VCOs. Given the consensus phase difference (c, s) , we can estimate (a, b) and (α, β) using

$$\begin{aligned}(\tilde{a}, \tilde{b}) &\approx (\alpha, \beta) \cdot \overline{(c, s)} \\ (\tilde{\alpha}, \tilde{\beta}) &\approx (a, b) \cdot (c, s).\end{aligned}$$

In other words, we rotate (a, b) clockwise to get $(\tilde{\alpha}, \tilde{\beta})$, an approximation of (α, β) . Then we can compute a phase correction,

$$(\Delta\alpha, \Delta\beta) = (\tilde{\alpha}, \tilde{\beta}) - (\alpha, \beta).$$

Likewise, we rotate (α, β) counter-clockwise to get $(\tilde{a}, \tilde{b})$, and compute a phase correction,

$$(\Delta a, \Delta b) = (\tilde{a}, \tilde{b}) - (a, b).$$

This process is illustrated in Fig. 11.

Only half of each correction needs to be incorporated to bring the two oscillators into the correct phase relationship. In our implementation, we divide the corrections by 5 and feed them back into the (x, y) components of the corresponding VCOs. The divisor of 5 was chosen instead of 2 for stability reasons. In our experience, a wide range of divisors work equally well.

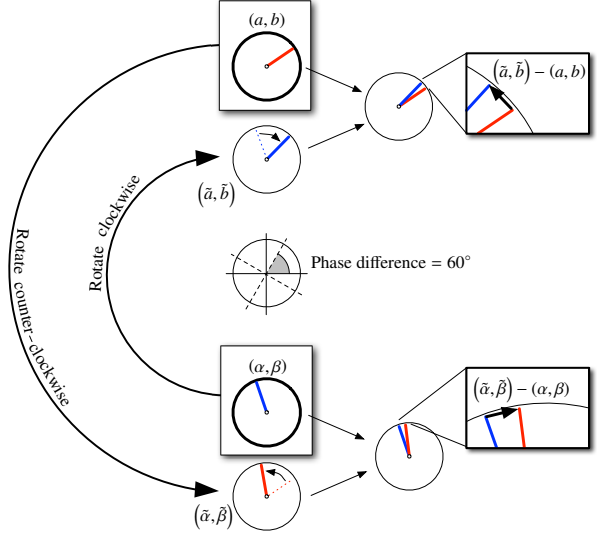


Figure 11: Calculation for coupling VCOs using phase-step nodes. In this figure, the consensus phase difference is 60° , as indicated in the centre of the figure. The VCO states are shown in the boxes labeled (a, b) and (α, β) . Each VCO is rotated into alignment (nominally) with the other VCO. The error vectors are fed back as corrections to the VCOs.

This phase-coupling method maintains a linear progression in phase across each propellor array of VCOs. However, there is nothing keeping the propellers in phase with each other. There are two forms of phase locking required to keep all the propellers consistent. Notice that all DC nodes should be in phase with each other since there is no direction-dependent frequency modulation on the DC nodes; only the speed affects their frequency. Hence, we need to make sure the DC components match across all the propellers. We achieved this by adding a single 6-D node to perform a coupling similar in nature to the phase-step coupling described above. This DC coupling node acts as a mechanism for finding a consensus phase among the DC nodes. This function could also be accomplished using random connections between DC nodes, similar to how the phase-step nodes arrive at a consensus.

A more complex form of coupling is required to

keep the 1-D phase ramps from the individual propellers coplanar with each other. While the phase-step nodes keep the phase linear within a 1-D propellor, we still need a way to ensure that the VCO phases form a linear function (a plane) in 2-D that passes through the origin. For example, drift could cause one propellor to attain a disproportionately steep slope that makes it tilt out of the plane delineated by the other propellers.

In order to ensure that the 1-D phase ramps stay linearly consistent, we need to couple together three phase-step nodes (from three different propellers). The resulting phase adjustments are fed back to the phase-step nodes.

3.4 Simulation of Rat Motion

We created our network model to test some specific aspects of the Fourier model. In particular, we wanted to see if we would find grid cells that fired spikes on a hexagonal grid of locations. We also wanted to see if these grid cells would exhibit phase precession compared to a global theta cycle. We added a 2-D VCO node that oscillates at approximately 8 Hz, and used this node’s state as the authoritative theta cycle.

To simulate the movement of a rat in a circular environment, we added to our model a random-walk function that adjusts the velocity vector smoothly. The resulting simulated rat trajectories are shown later. One could predict the rat’s location by numerically integrating the rat’s velocity. However, the rat’s own perceived location (as encoded in the phase ramp of the EC VCOs) soon drifted away from the computed position. This drift phenomenon has been observed before (Zilli and Hasselmo, 2010) and is probably due to temporal delays in network activity, network transients caused by sudden input changes, and inaccuracies in the frequencies of the VCOs³. A real rat seems to avoid this problem by updating its perceived location with sensory information (Burgess et al., 2007). Our model has no sensory input (though

we would consider it for future work). In terms of assessing the spatial maps of the rat’s EC neurons, what is important is where the rat believes it is, not necessarily where the rat actually is (Blair et al., 2007; Barry et al., 2007). We determine the rat’s perceived location from the slopes of the phase ramps of the three propellers. In particular, the phase-step nodes encode the slope that we need⁴. Each propellor gives us a projection of the rat’s position onto the propellor. Combining the three projections gives us an over-determined system; we find the least-squares solution to get a good estimate of the rat’s perceived location.

3.5 Network Architecture

As shown in Fig. 12, the network consists of three “wheels” of nodes, along with a velocity node, DC phase-coupling node, a theta-cycle node, and an array of grid-cell nodes. Each wheel has three propellers at angles 0°, 120°, and 240° (though a full model would include more propellers per wheel). The first wheel contains 17 VCO populations per propellor. Each population has 300 LIF neurons and encodes a 3-D vector. The recurrent connections of these oscillating populations have a synaptic time constant (τ_s in (3)) of 10 ms.

The phase-step wheel also has 3 propellers, but with 16 nodes per propellor (since they model the phase differences between the VCO nodes). Each phase-step population has 500 LIF neurons and encodes a 6-D vector as described in the last section. The coplanar coupling wheel mirrors the phase-step wheel, with each coplanar coupling node having 500 LIF neurons and encoding a 6-D vector.

The grid-cell array has 17 nodes, mirroring the 17 nodes in each of the VCO propellers. Each grid node contains 200 LIF neurons and encodes a 2-D vector of the sum of the phase states from the three corresponding VCOs. That is, each grid node receives the phase state from a triad of VCOs and simply adds

³We did not assess the accuracy of the VCO frequencies after implementing the various couplings, though it would be an interesting study.

⁴Proof that the rat’s location is projected onto each propellor comes from yet another fascinating and useful property of the Fourier Transform, the Fourier Projection Theorem. This property is used heavily in computed tomography reconstruction.

them together.

The DC phase-coupler node has 500 LIF neurons and encodes a 6-D vector that duplicates the phases of the three DC nodes. The velocity node has 100 LIF neurons and encodes a 2-D vector. Finally, the theta-cycle node contains 500 LIF neurons and encodes a 2-D vector that oscillates at approximately 8 Hz. The recurrent connections of the theta-cycle population use a synaptic time constant of $\tau_s = 5$ ms.

Unless otherwise specified, we used the following parameter values for all neurons: synaptic time constant $\tau_s = 5$ ms, refractory period $\tau_{\text{ref}} = 2$ ms, membrane time constant $\tau_m = 20$ ms, spiking threshold $J_{\text{th}} = 1$, encoding vectors ($\mathbf{e}_n$ from (2)) selected randomly (uniformly) from the unit hyper-sphere, neural gain and bias (α_n and β_n from (2)) chosen to randomly (uniformly) sample the unit hyper-sphere of the representational space, with a maximum firing rate in the range 200-400 Hz.

4 Results

The simulations were run using the Nengo software package (nengo.ca). The whole model includes 119 nodes, for a total of 68,700 LIF neurons. We ran the model for 300 seconds simulation time. The execution of the model took about 110 minutes to run on a laptop with a 2.9GHz Intel Core i7 processor and 8GB of RAM.

4.1 Grid Cells

Figure 13 shows a sampling of grid cells, with their spikes superimposed overtop of the rat’s trajectory. In the figure, the Fourier frequency of the grid-cell triad increases from left to right. The red dots of spikes clearly occur on a hexagonal grid with different scales. Not all neurons in the grid nodes exhibited grid firing patterns. However, about 10% did.

4.2 Theta-Phase Precession

If we focus on the timing of the grid-cell spike bursts, we can see that the start of the bursts precess through the theta cycle. Figure 14 plots the spikes as red lines

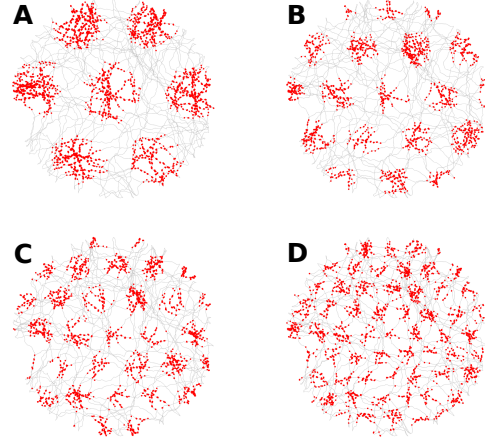


Figure 13: Spikes from grid cells superimposed on the rat’s trajectory. All the grid cells were taken from triads with an orientation of 0° . The neuron in A is from a grid-cell node at position 2 (where the central, or DC, grid-cell node is index 0). The neurons in B, C and D are from grid-cell nodes 3, 4 and 6, respectively.

over the theta cycle produced by the “theta” node. As Fig. 9 implies, the frequency of oscillation for the VCOs – and hence the grid cells – is higher than the nominal 8Hz theta cycle. Thus, we see the bursts of grid-cell activity precess through the lower-frequency theta cycle.

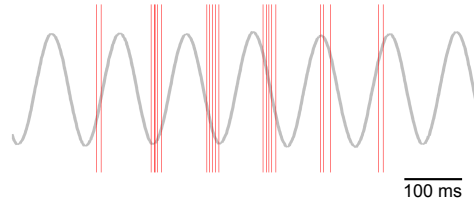


Figure 14: Theta-phase precession of grid-cell spikes.

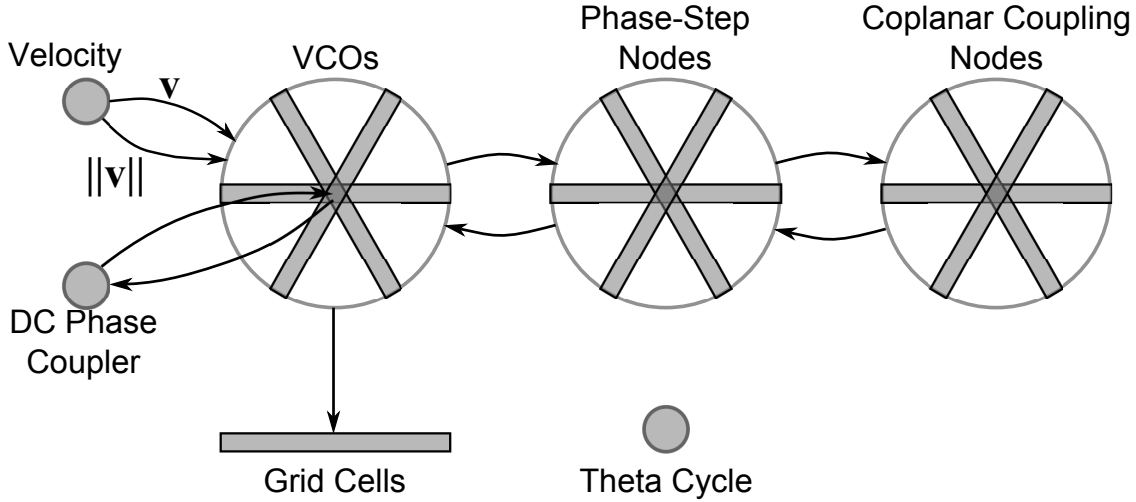


Figure 12: Network overview. The velocity node modulates the frequency of the VCOs (see equation (4)). The phase-step nodes couple the VCOs to maintain a 1-D phase ramp within each propellor. The coplanar coupling nodes further keep the phase slopes of the different propellers linearly consistent (so that they all rest in a common plane). The DC phase coupler node keeps the absolute phase of the propellers in sync. The grid cells sum triads of VCOs. The theta cycle node is a stand-alone oscillator that maintains a frequency of approximately 8Hz.

5 Discussion

The model proposed in this paper is similar to that proposed by Welday et al. (2011). However, our contribution is to re-formulate the pieces into a coherent and over-arching framework that allows for further analysis and deeper understanding. We propose that the EC acts as a bank of VCOs, organized in a polar fashion throughout the frequency domain. Movement of the animal induces frequency modulation of those VCOs according to where they rest in the frequency domain.

These VCOs project to spatial-map cells, such as place cells, grid cells, etc. The VCOs oscillate, so various combinations of them can result in complex interference patterns. Fourier Theory is the right tool to understanding these interference patterns. We propose that the connection strengths constitute the Fourier coefficients of the spatial map for the cell they project to. Each Fourier Transform implemented by the connections is modulated by the phase state of

the VCOs. Since the VCOs are constrained to maintain a linear phase ramp, the spatial maps get shifted. The slope of the phase ramp encodes the animal’s position in its environment, so the spatial maps are all shifted in concert with the animal’s motion. This framework makes it easy to understand the relationship between the VCOs and cells that exhibit spatial maps.

Grid cells might emerge as a by-product of a phase coupling mechanism. Some research has shown that the distributed nature of grid-cell encoding offers better accuracy than the same number of sparse place cells (Mathis and Herz, 2012). But this theory still does not address why grid cells appear, since the bank of VCOs also offers a distributed representation of location. Another theory, and one that we plan to investigate, is that grid cells are a by-product of the coupling mechanisms that maintain the phase relationships within the bank of VCOs. It seems intuitive that place cells could offer a stable and accurate representation of location as long as the underlying net-

work that feeds into the place cells encodes location in a stable and accurate manner. Coupling between nodes harnesses the redundancy in the network and enables resources to be focussed on lower-dimensional data, such as location. The coplanar-coupling nodes assess the linear consistency among three or more other nodes. In general, a linearity constraint in 2-D will always require input from at least three VCOs (in addition to the implicitly included DC node). We plan to investigate more general implementations of the coplanar constraint and observe whether these mechanisms inherently generate grid-cell behaviours.

Couching the behaviour of the EC in terms of Fourier Theory opens up a new vista of interpretations and predictions. It gives us the mathematical machinery to contemplate other neurophysiological observations. For example, how might sensory feedback be incorporated into the EC? If an animal is given a visual cue of its location, that sensory data might excite the corresponding place cell, which – in turn – could feed back through to the phase-step nodes to adjust the slope of the phase ramp so that it matches the animal’s location. It would seem that this feature would be accompanied by a phase-resetting mechanism that allowed the VCOs to rapidly realign their phases (or take on some other phase-ramp state).

The network we have built involves 119 populations, and contains a total of 67,800 LIF neurons. Our implementation is an important step in demonstrating the capabilities and behaviours of our model. However, an obvious question remains, how might such a system get established? What self-organizing principles might apply, and where? Spatial maps of place cells have been learned using Hebbian learning (Rolls et al., 2006). Grid cells can emerge spontaneously in a topographically connected network with local excitation and lateral inhibition (Fuhs and Touretzky, 2006; McNaughton et al., 2006). However, these “Turing grids” are not found in adults, leading researchers to suggest that they form during a developmental stage and are used to guide the formation of grid cells in the non-topographical, adult EC network. Even a random selection of grid cells can produce place cells (Solstad et al., 2006; de Almeida et al., 2009). We plan to investigate unsupervised

and supervised learning algorithms to derive neural oscillators. One could also look at how such oscillators could take on the proper phase coupling.

The concept of using frequency modulation to encode spatial information is used in magnetic resonance imaging (MRI). MRI is a medical imaging modality that causes nuclear dipoles in tissues to precess, or oscillate. The MRI scanner is only capable of sensing the sum of these spinning dipoles, so one might think that there is no way to get spatial information from this sum. However, the scanner can cause linear gradients in the oscillation frequency, causing a phase ramp in the dipoles. In essence, MRI raw data is encoded in the frequency domain. These phase ramps allow the scanner to sample different Fourier coefficients, and build a picture of the anatomy by taking the IDFT of that data.⁵ This process is often called “spatial coding” or “phase encoding”.

6 Conclusion

Although a number of theories have been forwarded regarding the relationships between place cells, grid cells, phase precession, and other spatial-map cells, none have explained all the components with a single overarching framework. Our Fourier model of the entorhinal cortex path integration system organizes the pieces into an architecture with a rich and well-understood foundation. Knowledge about other properties of the Fourier Transform can help to guide further development of the model, and assess how it may (or may not) be extended to explain or predict other observations.

Acknowledgement

We are thankful for the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Foundation for Innovation (CFI), and the Ontario Innovation Trust.

⁵The Fourier Projection Theorem is also used in MRI.

References

- Barry, C., R. Hayman, N. Burgess, and K. Jeffery (2007). Experience-dependent rescaling of entorhinal grids. *Nature Neuroscience* 10(6), 682–684.
- Blair, H., K. Gupta, and K. Zhang (2008). Conversion of a phase- to a rate-coded position signal by a three-stage model of theta cells, grid cells, and place cells. *Hippocampus* 18(12), 1239–1255.
- Blair, H., A. C. Welday, and K. Zhang (2007, March). Scale-Invariant Memory Representations Emerge from Moire Interference between Grid Fields That Produce Theta Oscillations: A Computational Model. *Journal of Neuroscience* 27(12), 3211–3229.
- Burgess, N., C. Barry, and J. O’Keefe (2007). An oscillatory interference model of grid cell firing. *Hippocampus* 17(9), 801–812.
- Conklin, J. and C. Eliasmith (2005). A Controlled Attractor Network Model of Path Integration in the Rat. *Journal of Computational Neuroscience* 18, 183–203.
- de Almeida, L., M. Idiart, and J. E. Lisman (2009). The input–output transformation of the hippocampal granule cells: from grid cells to place fields. *The Journal of Neuroscience* 29(23), 7504–7512.
- Eliasmith, C. and C. H. Anderson (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. Cambridge, MA: MIT Press.
- Eliasmith, C., T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen (2013). A large-scale model of the functioning brain. *Science*.
- Fuhs, M. C. and D. S. Touretzky (2006). A spin glass model of path integration in rat medial entorhinal cortex. *The Journal of Neuroscience* 26(16), 4266–4276.
- Hafting, T., M. Fyhn, S. Molden, M. Moser, and E. Moser (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature* 436(7052), 801–806.
- Koch, C. (1999). *Simplified models of individual neurons*. Biophysics of computation.
- Mathis, A. and A. Herz (2012). Optimal Population Codes for Space: Grid Cells Outperform Place Cells. *Neural Computation* 24, 2280–2317.
- McNaughton, B., F. P. Battaglia, O. Jensen, E. Moser, and M. Moser (2006). Path integration and the neural basis of the ‘cognitive map’. *Nature Reviews Neuroscience* 7(8), 663–678.
- Moser, E., E. Kropff, and M. Moser (2008). Place cells, grid cells, and the brain’s spatial representation system. *Annu. Rev. Neurosci.* 31, 69–89.
- O’Keefe, J. and N. Burgess (2005). Dual phase and rate coding in hippocampal place cells: theoretical significance and relationship to entorhinal grid cells. *Hippocampus* 15(7), 853–866.
- Oppenheim, A. V. and R. W. Schaffer (1999). *Discrete-Time Signal Processing* (Second ed.). Prentice Hall.
- Rolls, E. T., S. M. Stringer, and T. Elliot (2006, January). Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning. *Network: Computation in neural systems* 17(4), 447–465.
- Solstad, T., E. Moser, and G. T. Einevoll (2006). From grid cells to place cells: a mathematical model. *Hippocampus* 16(12), 1026–1031.
- Welday, A. C., I. G. Shlifer, M. L. Bloom, K. Zhang, and H. T. Blair (2011, November). Cosine Directional Tuning of Theta Cell Burst Frequencies: Evidence for Spatial Coding by Oscillatory Interference. *Journal of Neuroscience* 31(45), 16157–16176.
- Zilli, E. A. (2012, April). Models of grid cell spatial firing published 2005–2011. *Frontiers in Neural Circuits* 6(16), 1–17.
- Zilli, E. A. and M. E. Hasselmo (2010, October). Coupled Noisy Spiking Neurons as Velocity-Controlled Oscillators in a Model of Grid Cell Spatial Firing. *Journal of Neuroscience* 30(41), 13850–13860.