Oscillatory neurocomputing with ring attractors: A network architecture for mapping locations in space onto patterns of neural synchrony

Hugh T. Blair¹, Allan Wu², and Jason Cong²
¹UCLA Psychology Department, Los Angeles, CA 90095; ²UCLA Computer Science Department, Los Angeles, CA 90095

Supplementary Material

Gaussian weight vectors for the ring attractor circuit

As in Song and Wang [51], the excitatory and inhibitory layers are reciprocally interconnected with themselves and one another by Gaussian weight vectors,

\[ w_{i,j} = \alpha \times \exp \left( \frac{\cos \left( \rho(i) - \rho(j) - \mu \right)}{\sigma^2} \right), \]  

where \( i \) and \( j \) index neurons in the source and target layers, respectively, \( \alpha \) is a scaling factor for the weights, \( \mu \) is referred to as the center phase of the connection between the source and target layers, \( \sigma \) is the angular width (standard deviation in radians) of the weight vector, and \( \rho(i) \approx 2\pi \frac{i}{M} \) is the phase angle (in radians) assigned to the neuron at position \( i \) in the ring layer. For simulations presented here, there were \( M = 108 \) neurons in each layer, for a total of \( 2M = 216 \) neurons in both layers of each ring. There were no recurrent connections in the excitatory layer, so \( \alpha = 0 \) for recurrent excitatory weights. The recurrent inhibitory weight vector and the projection from the excitatory to inhibitory layer had zero center phases (that is, \( \mu = 0 \)), whereas the projection from the inhibitory to excitatory layer had a center phase of \( \mu = 2\pi / 3 \) to induce rotational asymmetry in the connections between layers, so that the activity bump constantly circulated in the clockwise direction at a frequency that was regulated by velocity inputs.

Dependence of spatial reference phase upon \( \mu \)

As activity bumps circulated around the three rings, they drove three different oscillatory conductances, \( g_1, g_2, g_3 \), in the postsynaptic grid cell. The grid cell fired when these three conductances were synchronized; that is, when excitatory conductances were in phase with one another and in antiphase with the inhibitory conductance. Each \( g_n \) was maximal at moments when \( \phi_n(t) = \mu_n \) (that is, when the activity bump passed through the center phase of the grid cell’s input from ring \( n \)). The maximal conductance either depolarized or hyperpolarized the grid cell (depending upon whether the grid cell’s inputs from ring \( n \) were excitatory or
inhibitory), so from the grid cell’s vantage point, $\mu_n$ can be equated directly with the spatial reference phase $\varphi_n$ as follows:

$$
\varphi_n = \begin{cases} 
\mu_n, & \text{for excitatory inputs} \\
\mu_n + \pi, & \text{for inhibitory inputs}
\end{cases}
$$

(S3)

This relation between $\mu_n$ and $\varphi_n$ defines how center phases of the weight vectors determine when (on what phase of the theta cycle) and where (at what spatial location) the grid cell fires.

*Decoding phase from spike trains*

The ring oscillator’s instantaneous phase, $\phi_k(t)$, was estimated from spike trains generated by neurons in the ring. Spikes were sampled in successive, non-overlapping time windows of the simulation; the sample window had to be narrow enough to approximate an instantaneous phase from the spikes, but wide enough to reliably obtain at least one spike (and preferably more) per sample window. These competing constraints were better satisfied by spike trains from the inhibitory layer than the excitatory layer, because inhibitory cells had higher firing rates. Hence, the ring’s phase was estimated by circular averaging of inhibitory cell spike phases from 10 ms sample windows taken throughout the simulation:

$$
\phi(t) \approx \text{atan2} \left( \sum_{k=1}^{K} \sin \rho(i_k) K, \sum_{k=1}^{K} \cos \rho(i_k) K \right) + \xi^R(t),
$$

(S4)

where $K$ denotes total the number of spikes recorded in the 10 ms sample window, $i_k$ is the ring position of the cell that fired the $k^{th}$ spike, and $\rho$ is defined by Eq. (S2).

*Phase resetting analysis and feedback weight vectors*

Phase resetting behavior of the ring oscillator circuit was tested by delivering a Gaussian-weighted entrainment pulse at different offsets from the center of the circulating activity bump. The entrainment pulse consisted of two excitatory input spikes spaced 4 ms apart, delivered simultaneously to all theta cells in the excitatory (but not the inhibitory) layer of the ring. In simulations, the center phase was assigned a different value every time the reset pulse was delivered by setting $\mu = \phi(t) + \Delta \phi(t)$, where $\phi(t)$ denotes the ring oscillator’s decoded phase at that moment of pulse delivery, and $\Delta \phi(t)$ was randomly assigned different values on each trial, ranging between $-\pi$ and $+\pi$ in increments of $\pi/6$. Under this arrangement, the entrainment pulse led the activity bump when $0 > \Delta \phi(t) > \pi$, lagged the activity bump when $2\pi < \Delta \phi(t) < \pi$, overlapped with the center the activity bump when $\Delta \phi(t) = 0$, and was evenly balanced between lead and lag when $\Delta \phi(t) = \pi$ (Fig. S1). The phase shift induced by the reset pulse was measured by allowing the ongoing simulation to pass through one full cycle of an oscillation (that is, one full circuit of the activity bump around the ring) with no reset pulse, and second by restoring saved state variables so that the network was returned to exactly the same state it had been in at the start of the completed cycle period; the simulation was then re-run with the first spike of a reset pulse delivered on the first time step of the restarted simulation. The length of the cycle period was compared for the first and second iterations of the simulation, yielding two period measurements, $P_1$ and $P_2$, so that the phase shift induced by the reset pulse could then be
quantified as a period change, \( \Delta P = 2\pi(P_2 - P_1) / P_1 \). Three independent sets of simulations were conducted, in which the ring oscillator’s activity bump circulated at three different mean frequencies (approximately 6.7, 7.4, and 8.2 Hz). Phase resetting curves (PRCs) exhibited zero crossings with negative slope. The average effect of the entrainment pulse was to push the activity bump towards a phase position \( \sim \pi / 6 \) radians ahead of its current position (Fig. S1). Hence, on average, a ring oscillator’s phase was unaffected when a Gaussian-weighted input pulse to the ring was centered at a position \( \pi / 6 \) radians in front of the activity bump’s current position, and the ring’s phase was advanced or delayed, respectively, by inputs arriving ahead of or behind the neutral offset at \( \pi / 6 \). This “neutral” value of the phase shift parameter at \( \Delta \phi \approx \pi / 6 \) was exploited to correct phase errors via feedback to the rings from grid cells. To achieve this, the center phases for feedback projections were set to \( \varphi_n + \pi / 6 \); that is, the feedback center phase was obtained by adding \( \pi / 6 \) to the feedforward center phase. Consequently, when a grid cell fired, the system was nudged toward an attractor at \( \tilde{\phi} = \tilde{\varphi} \).

Grid cell current and conductance equations

Each grid cell received input from three ring oscillators, originating from the inhibitory layer of one ring and the excitatory layer of the other two rings. The total synaptic current to the grid cell was thus expressible as the sum of one excitatory and two inhibitory currents:

\[
I(t) = g^{GABA}(t)\left[ E^{GABA} - V(t) \right] + \sum_{n=1}^{2} g^{GLU}_n(t)\left[ E^{GLU}_n - V(t) \right],
\]

where \( V(t) \) is the grid cell’s membrane potential, \( E^{GABA} \) and \( E^{GLU} \) are the reversal potentials for GABA and glutamate synapses, respectively. Each input conductance was updated by computing a leaky integral of weighted spike train inputs from neurons in the source layer,

\[
g(t) = -\tau g(t-1) + \sum_{i=1}^{M} \delta_i(t)w_i,
\]

where \( \tau \) is the conductance decay time constant, \( w_i \) is a Gaussian vector like that defined in Eq. (S1), and \( \delta_i(t) \) is a delta function that signals the occurrence of input spikes by adopting the value \( \delta_i(t) = 1 \) if a spike was fired by cell \( i \) of the source layer at time \( t \), and \( \delta_i(t) = 0 \) otherwise. Since Eq. (S6) describes triggering of a single composite conductance (rather than a vector of individual synaptic conductances) by the source layer (see Methods), the term for the target phase in Eq. (S1) may be dropped by setting \( \rho(j) = 0 \); the weight vector’s center phase then determines the value of the spatial phase parameter, \( \varphi_n \), in Eqs. (6) and (7) of the main text.

Feedforward grid cell weight vectors

The grid module circuit (Fig. 4A) contains three rhomboidal sheets of grid cells, with \( P \) cells along the length of each side (hence, there are \( P^2 \) grid cells in each sheet, and \( 3P^2 \) grid cells in the full grid module; \( P = 6 \) in simulations presented here). Grid cells within a sheet have differing translational phases, evenly distributed over the rhombus-shaped domain, \( \tilde{D} \) (see Fig. 3B). Grid cells residing at the same position in different sheets have the same translational phase,
so center phases of the weight vectors were chosen to satisfy \( \mathbf{x}(\mathbf{\Phi}^{p,q}_1) = \mathbf{x}(\mathbf{\Phi}^{p,q}_2) = \mathbf{x}(\mathbf{\Phi}^{p,q}_3) \), where subscripts index the three sheets, and superscripts \( p \) and \( q \) denote the row and column position, respectively, of a grid cell within its own sheet. Synaptic weights of connections from ring oscillators to grid cells were chosen to satisfy the constraint that adjacent grid cells in each sheet had grid fields with adjacent translational phases, and also to obey the convention that each grid cell received the inhibitory layer of one ring and the excitatory layer of the other two rings [see Eq. (S5)]. To satisfy all of these constraints, the center phases for the feedforward weight vectors were chosen in such a way that the local reference phase vectors for grid cells at each position in the three sheets acquired the following values after applying Eq. (S3):

\[
\varphi^{p,q}_1 = \frac{2\pi}{P} \begin{bmatrix} 0 \\ \frac{p-1}{3} \\ \frac{p-q}{3} \end{bmatrix}, \quad \varphi^{p,q}_2 = \frac{2\pi}{P} \begin{bmatrix} 1-p \\ 0 \\ 1-q \end{bmatrix}, \quad \varphi^{p,q}_3 = \frac{2\pi}{P} \begin{bmatrix} q-p \\ q-1 \\ 0 \end{bmatrix} + \frac{2\pi}{3}.
\]

Each local reference phase vector contains three elements. We may denote each element by \( \varphi^{p,q}_{j,k} \), where \( k \) indexes the grid sheet and \( j \) indexes the input ring. To satisfy Eq. (S5), we adopt the convention that all of the grid cells in sheet \( k \) receive input from the inhibitory layer of the ring indexed by \( j = k \), and from the excitatory layers of the other two rings (those for which \( j \neq k \)). Eq. (S7) sets \( \varphi^{p,q}_{j,k} = 0 \) at all entries where \( j = k \), and from this it follows that all of the inhibitory inputs to ring \( k \) are perfectly synchronized with one another (because they share the same zero center phase). This has the effect of synchronizing theta modulation of the grid cells across the entire sheet, because grid cells can only fire when their two excitatory inputs from rings \( j \neq k \) are in antiphase with their the inhibitory inputs from ring \( j = k \). The conditions under which this occurs are determined solely by the values of the nonzero entries in the column vectors of Eq. (S7), which differ for each grid cell depending upon its position \( (p, q) \) within the sheet. The addition of a constant (in this case, \( \pm 2\pi / 3 \)) to the local reference phase vector shifts the temporal phase of the grid cell’s modulation by theta rhythm, without affecting the spatial phase of its firing fields. Consequently, Eq. (S7) specifies weight assignments for which grid cells in the same sheet fire on the same phase of the theta cycle (because of their synchronized inhibitory input), and grid cells residing in different sheets fire on different phases of the theta cycle (because a different constant is added to their local reference phase vectors).

**Simulations of path integration error accumulation**

The rate of error accumulation was quantified by running multiple simulations like those shown in Fig. 4, and measuring the mean time delay, \( \tau_x \), until one of the two synchronization vector elements deviated by \( \pi \) or more from its initial value; that is, until \( \tilde{\varphi}_1(t) \geq \tilde{\varphi}_1(0) \pm \pi \) or \( \tilde{\varphi}_2(t) \geq \tilde{\varphi}_2(0) \pm \pi \). This is a sensible divergence criterion since \( \tau_x \) marks the time at which the decoded position no longer lie within the same rhombus, \( \tilde{D} \), as the true position. The rate of error accumulation depended upon the base frequency, but in general, \( \tau_x \) was measured on a
time scale of seconds. It was found that $\tau_\pi \approx 4.8 \pm 0.8$ for $T = 15$ KHz, $\tau_\pi \approx 5.6 \pm 1.2$ for $T = 20$ KHz, and $\tau_\pi \approx 6.1 \pm 1.4$ for $T = 25$ KHz (averaging over 30 simulations for each $T$).

Model implementation

Simulations were implemented in C++ on UCLA’s Hoffman2 computing cluster. The model was developed in conjunction with ongoing efforts by our group to create high-speed, low-power, and lightweight hardware platforms for spiking neural network models, and for this reason, all parameters and state variables were stored as fixed-point integers to facilitate implementations on field-programmable gate array (FPGA) chips (Blair et al., 2013). Execution time was optimized by selecting simulation parameter values so that arithmetic operations such as multiplication and division could be implemented as bit shift operations whenever it was possible to do so without compromising biological realism (see [54]). The model contained four distinct types of neurons (Fig. 6A): inhibitory ($inh$) and excitatory ($exc$) ring oscillator neurons, grid cells ($grid$), and velocity cells ($vel$). Output from $vel$ neurons was represented by stochastic spike trains, so voltage and conductance were not numerically integrated for $vel$ neurons during the simulation. The $inh$, $exc$, and $grid$ populations were simulated as leaky integrate-and-fire neurons. Numerical integration was performed using the forward Euler method with an integration time step of $dt = 1/25$ ms. For simplicity, axonal conductance delays were not simulated and thus equal to zero.

Voltage integration

Membrane voltages, $V$, were represented as 16-bit unsigned integers. Fixed point conversion of $V$ was achieved by mapping a membrane voltage range of -70 to +30 mV onto the unsigned integers $0 \leq V \leq 65535$. Each neuron’s membrane voltage was held at a fixed reset potential, $V_{reset}$, during time steps when it was in the post-spike refractory state. At all other time steps, $V$ was updated by numeric integration using the forward Euler method:

$$V(t+1) = \begin{cases} V_{reset}, & \text{if } R(t+1) > 0 \text{ (post-spike refractory period)} \\ [(V(t)+I_{in}(t)-I_{out}(t))\left(\frac{V(t)+I_{in}(t)}{V_{thresh}}\right) > I_{out}(t)], & \text{otherwise} \end{cases}$$ (S8)

where $t$ is the simulation time step, $I_{in}$ and $I_{out}$ are the total inward (excitatory) and outward (inhibitory) membrane currents, and $R$ is the value of the neuron’s refractory period clock, updated prior to $V$ at each time step as follows:

$$R(t+1) = R(t) + T_{refract}[V(t) > V_{thresh}] - 1,$$ (S9)

where $V_{thresh}$ is the spike threshold and $T_{refract}$ is the number of time steps in the refractory period. In both Eqs. (S8) and (S9), [] denotes Iverson brackets.

Current equations

The total membrane current at each time step was a sum of inward and outward synaptic currents, plus a leak current:

$$I_{total}(t) = (I_{in}(t) + I_{out}(t) + I_{leak}(t)) \gg S_{dt},$$ (S10)
where $\gg S_{dt}$ is the rightward bitshift operator in C++. which simultaneously implements both multiplication by the integration time step ($dt = 1/25$ ms) and division by the membrane capacitance, $C_m$. In simulations presented here, $S_{dt}$ was set to values corresponding with $C_m = 0.32$ nF for postsynaptic exc and grid cells, and $C_m = 0.16$ nF for postsynaptic inh cells.

The inward (excitatory) synaptic current for each neuron in the model was computed by summing over three sources, which corresponded to the three types of excitatory neurons:

$$I_{in}(t) = I_{ext}(t) + I_{exc}(t) + I_{grid}(t) = F_{ext}(t)g_{ext}(t) + F_{exc}(t)g_{exc}(t) + F_{grid}(t)g_{grid}(t).$$  \hspace{1cm} (S11)

The component currents ($I_{ext}$, $I_{exc}$, and $I_{grid}$) represent the sum total current generated in a postsynaptic cell by all weighted inputs from neurons in each excitatory population (one or more of these components can be zero if a neuron does not receive any input from the corresponding population). Each component current is computed as the product of a population conductance, $g$, which shall be defined below, and an inward driving force, $F$, computed by

$$F_{ext}(t) = (V(t) - E_{AMPA}) \gg S_{ext},$$
$$F_{exc}(t) = (V(t) - E_{NMDA}) \gg S_{exc},$$
$$F_{grid}(t) = (V(t) - E_{AMPA}) \gg S_{grid},$$

where $E_{AMPA} = E_{NMDA} = 45875$ represent a 0 mV reversal potential for glutamatergic synapses, and $\gg$ is a rightward shift by a number of bits ($S_{ext}$, $S_{exc}$, $S_{grid}$) that modulates the gain of inputs from each excitatory population, and thus modulates the value of $\alpha$ in Eq. (S1). In simulations presented here, $S_{ext} = 12$ and $S_{grid} = 8$ for postsynaptic exc cells, $S_{exc} = S_{grid} = 8$ for postsynaptic inh cells, and $S_{exc} = 9$ for postsynaptic grid cells. The three driving forces in Eq. (S12) are always inward because $E_{AMPA} = E_{NMDA} > V_{thresh} > V_{reset}$, so $E - V(t)$ cannot be negative.

The outward (inhibitory) synaptic current was equivalent to the current generated by inputs from the population of inhibitory neurons:

$$I_{out}(t) = I_{inh}(t) = F_{inh}(t)g_{inh}(t).$$  \hspace{1cm} (S13)

where

$$F_{inh}(t) = V(t) \gg S_{inh}. $$  \hspace{1cm} (S14)

There is no reversal potential term in this expression because it was assumed that $E_{GABA} = 0$ in fixed integer units (corresponding to a value of -70 in units of mV). The rightward shift by $S_{inh}$ served as a global gain factor that modulates the value of $\alpha$ for inhibitory weights in Eq. (S1). For simulations presented here, $S_{inh} = 10$ for postsynaptic exc cells, $S_{inh} = 8$ for postsynaptic inh cells, and $S_{inh} = 10$ for postsynaptic grid cells.

The leak current was computed as

$$I_{leak}(t) = V(t) - E_{leak}. $$  \hspace{1cm} (S15)

where $E_{leak} = 0$ (corresponding to -70 mV) for postsynaptic exc and inh cells, and $E_{leak} = 6553$ (corresponding to -60 mV) for postsynaptic grid cells.

**Conductance parameters**
A separate ‘population conductance’ value was integrated in each postsynaptic neuron to represent summed inputs from each presynaptic population [51]. Population conductances for each type of input were computed by integrating a weighted sum over all inputs of that type, as described by Eq. (S6). Conductance parameters were chosen in such a way that multiplication by the decay time constant in Eq. (S6) could be implemented as a leftward integer bitshift operation. Decay time constants were 1.28 ms for AMPA and GABA conductances, and 20.48 ms for NMDA conductances. Voltage sensitivity of the NMDA channel was not simulated, because similar oscillatory dynamics were achievable with or without voltage sensitivity of the NMDA receptors. Hence, AMPA, NMDA, and GABA receptors were simulated by integrating the same conductance formula in Eq. (S6), differing only in their decay constants and weight vectors.

Weight vector parameters

Parameters of the Gaussian weight vectors differed for each combination of source and target populations. The width parameters were $\sigma = 2\pi / 3$ for connections from $exc$ to $inh$ neurons, $\sigma = 10\pi / 9$ for recurrent connections between $inh$ and $inh$ neurons, $\sigma = 10\pi / 9$ for connections from $inh$ to $exc$ neurons, and $\sigma = \pi / 12$ for feedback connections from $grid$ to $exc$ neurons. The scaling parameters were $\alpha = 463 / 108$ for connections from $exc$ to $inh$ neurons, $\alpha = 2500 / 108$ for recurrent connections between $inh$ and $inh$ neurons, $\alpha = 252 / 108$ for connections from $inh$ to $exc$ neurons, and $\sigma = 200 / 108$ for feedback connections from $grid$ to $exc$ neurons. The weight vector was stored as a vector of unsigned integers, and since values less than one cannot be represented in this format, some weight vectors were stored as by the vector of values $1 / w_i$, and multiplication by the weight vector in Eq. (S6) was implemented as unsigned integer division by the stored weight values, rather than as multiplication.