Path Integration using Coupled Bump Attractors

Arash Sal Moslehian and Ludwig Tiston

***Abstract*—Ants can return straight to their nest using an internal 'memory' of their location. This study will create a circuit to track ant movement with Poisson neurons and bump attractors, simulating and decoding their trajectory.**

***Index Terms*—Computational Neuroscience, Bump Attractor**

Ex.0.1: We can see in Fig. 1 a) As β increases (keeping $\alpha = 2$ constant), the function is shifted to the right and as it decreases the function moves to the left. In Fig. 1 b) As α increases (keeping $\beta = 0.5$ constant), the transition gets steeper and as it decreases, the transition gets less steep.

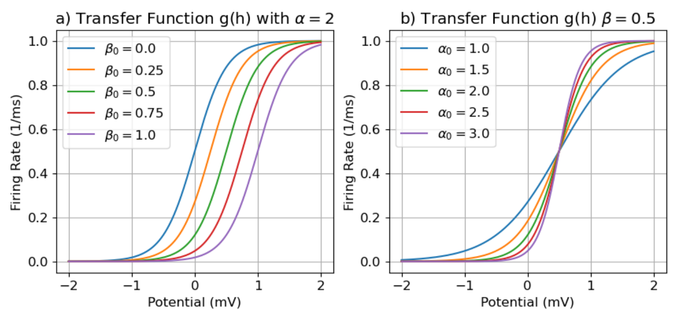


Fig. 1: The transfer function g as a funciton of the potential h. a) Changing β while keeping α constant. b) Changing α while keeping β constant.

Ex.0.2: Fig. 2 a) compares the mean number of spikes per ms across the N neurons to the theoretical instantaneous rate. We can see that the simulated firing rate is slightly delayed compared to the theoretical rate and has more fluctuations. As we increase the number of neurons to $N = 1000$ in Fig. 2 b) we can see the simulated firing rate is smoother and closer to the instantaneous rate. The fluctuations arise from the limited amount of neurons, as they increase the better approximate this mean-field limit. The delay stems from the timeconstant/scale which the instantaneous rate does not consider.

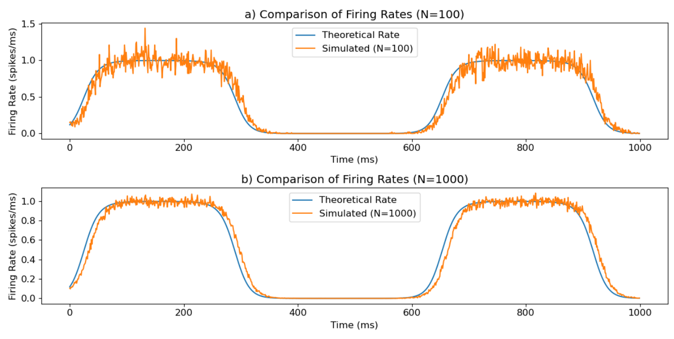


Fig. 2: Comparinson of simulated firing rate and the theoretical rate. a) With $N = 100$ neurons b) With $N = 1000$ neurons.

Arash Sal Moslehian is with EPFL, Lausanne:e-mail: arash.salmoslehian@epfl.ch
Ludwig Tiston is with EPFL, Lausanne:e-mail: ludwig.tiston@epfl.ch

Ex.1.1: In Fig. 3 a) shows the raster plot of the recurrent network with no external input. The initial conditions are sampled from $h_i(0)\ Uniform(0, 1)$ mV. Fig. 3 b) shows the mean firing rate of all the neurons for different values of J. For $J > 4.7$ we can consistently get a bump.

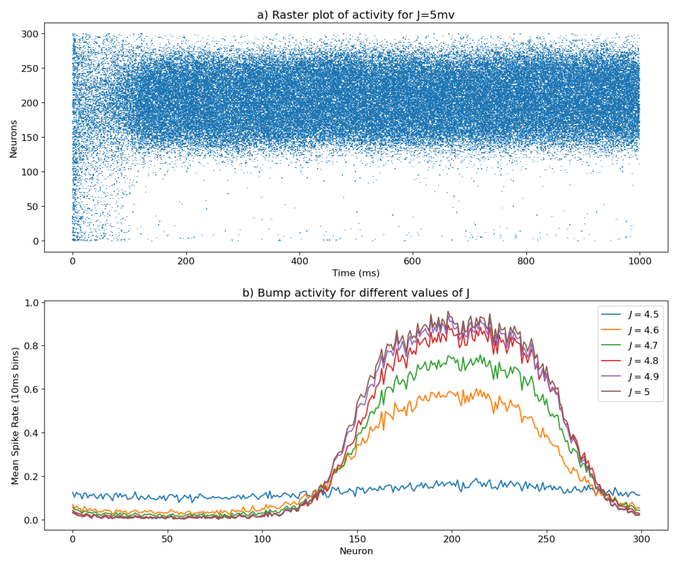
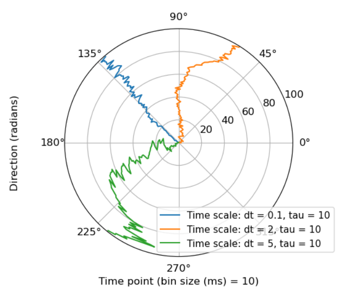
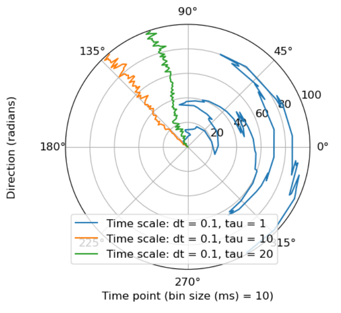


Fig. 3

Ex.1.2: Fig. 4 shows the location of θ_{bump} for different values of dt. As we increase the time scale we see the bump drift and move around a bit.



(a) Bump location for different values of dt



(b) Bump location for different values of τ

Fig. 4: Location of θ_{bump} over time for different time scale.

Ex.1.3: The drift in the location of the bump is primarily caused by the discrete nature and finite size of the network. Increasing N can provide a more continues approximation of the underlying dynamics and reduce the drift. Smaller dt makes the integration more accurate and causes the network to drift less. A smaller timeconstant τ amplifies these discrete fluctuations further. As seen in Fig. 4 larger time scales cause more drift.

Ex.1.4: Fig. 5 shows the result of stimulating the network using the gaussian inputs. These input create transient increase in the firing rate of neurons that are around the centers of the inputs. We see increased activity around the mean of the inputs during the time that they are on. The recurrent connections will help sustain the bump of activity for some time even after the external input is removed. The cosine connectivity profile means that neurons that are spatially close are more strongly connected. Therefore, an increase in activity at the center of the input will spread to neighboring neurons due to the recurrent connections.

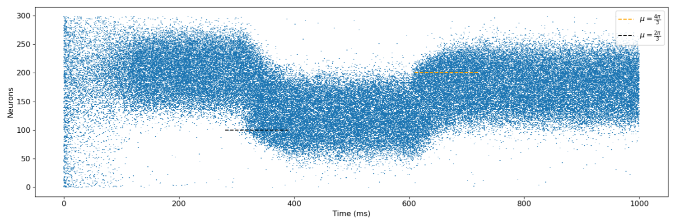
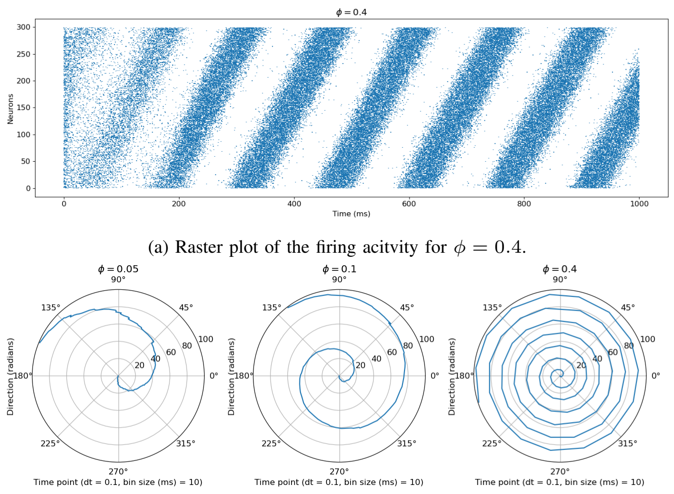


Fig. 5: Raster plot of the spikes with external input. The dashed lines outline the mean of the input gaussian.

Ex.1.5: In Fig. 6 we see that when a small angle bias is added to the connectivity profile, the bumps position shifts/rotates over time. As ϕ increases, the frequency of rotation increased. This is as expected, as the neurons are now stimulated at all times as if there was a bump slightly shifted from it, so it tried to "chase the shadow" in each timestep.



(a) Raster plot of the firing acitvity for $\phi = 0.4$.

(b) θ_{bump} for $\phi = 0.05$. (c) θ_{bump} for $\phi = 0.1$. (d) θ_{bump} for $\phi = 0.4$.

Fig. 6: Raster plot of the spikes with a biased angle ϕ in connectivity profile.

Ex.1.6: Fig. 7 shows the raster plot for bump attractor model using the gaussian tuning curve for $N = 100$ neurons and $T = 300$ ms as the simulation is computationally intensive. We also added a small $\phi = 0.2$ like the previous case to easier see the behaviour of the bump near the edges. From the lectures we know we want a connectivity profile that is like a "mexican hat". That is exitation close, inhibiton further away, and close to no interaction even further. The values $J_0 = -2, J_1 = 4.5, \sigma = 1$ fit this description and does generate a stable bump. The periodic nature of the ring model eliminates any boundary effect. For the line model, the two ends of the simulation space are not connected as in the the ring model. This has the effect that the bump gets stuck as the edge, instead of wrapping around.The bump cannot move further to the edge as there aren't enough neurons to stimulate it in that direction. We can see that with a small ϕ the bump is stuck at the edge and does not wrap around like in Fig. 6.

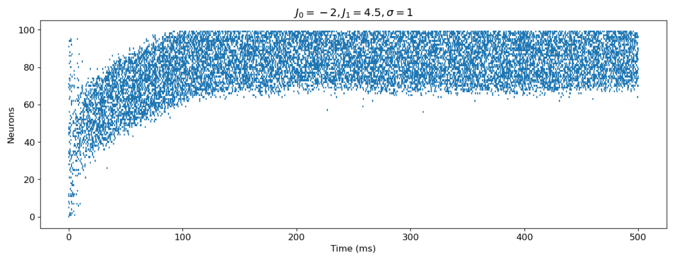


Fig. 7: Bump attractor model using gaussian tuning curve.

Ex.2.1: Equation 1 shows the equation of the full input received for each neuron in both populations. The push-pull system is formed as the result of two opposing "forces" in balance, acting on each population/bump. The phase shift θ is pushing it in one direction, but the stimulation from the other neuron population is pulling it in the other direction (through the weights $w_{R\to L}$ and $w_{L\to R}$).

When one population is externally excited it pushes the neurons in its population by exciting them and pulls on the neurons in the other population by inhibiting them. Also, when its bump moves in a certain direction, and the other population experiences an inhibitory effect that pulls its bump in the opposite direction. This mutual inhibition and excitation dynamic results in a stable configuration where the network can robustly integrate and represent a variable.

$$
\begin{aligned}
I_{left,i} &= \frac{J}{N}(\sum_{j=1}^{N} w_{L\to L}(x_i^L, x_j^L)S_j(t) \\
&\quad + \sum_{j=1}^{N} w_{R\to L}(x_i^L, x_j^R)S_j(t)) + I_{ext,i} \\
I_{left,i} &= \frac{J}{N}(\sum_{j=1}^{N} cos(x_i^L + \theta - x_j^L)S_j(t) \\
&\quad + \sum_{j=1}^{N} cos(x_i^L + \theta - x_j^R)S_j(t)) + I_{ext,i} \\
I_{left,i} &= J((cos(x_i^L + \theta)m_{cos_L}(t) + sin(x_i^L + \theta)m_{sin_L}(t)) \\
&\quad + (cos(x_i^L + \theta)m_{cos_R}(t) + sin(x_i^L + \theta)m_{sin_R}(t))) + I_{ext,i} \\
I_{right,i} &= J((cos(x_i^R - \theta)m_{cos_R}(t) + sin(x_i^R - \theta)m_{sin_R}(t)) \\
&\quad + (cos(x_i^R - \theta)m_{cos_L}(t) + sin(x_i^R - \theta)m_{sin_L}(t))) + I_{ext,i}
\end{aligned} \tag{1}
$$

Ex.2.2: Fig. 8 shows the location of the bump for the left and right population with randomly initialized potentials and $\theta = 10°$ and $J = 3$ pC.

Ex.2.3: The bump-position of the two populations, and their mean, can be seen in Fig. 9. Their mean has been initialised at π This is done by initialising the two membrane potentials

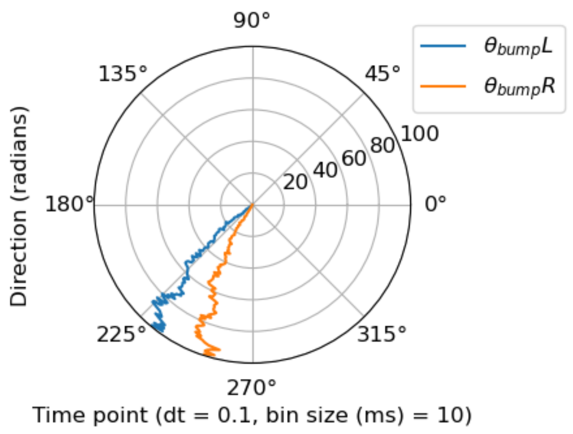


Fig. 8: Bump location for the coupled attractor.

with noisy square-waves slightly offset from π. A slight drift can be seen over time.

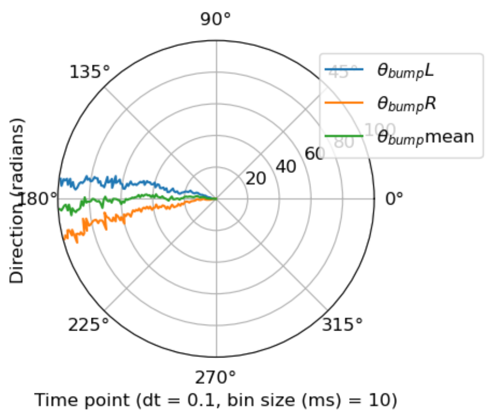


Fig. 9: Coupled attractor with mean location at π

Ex.2.4: Fig. 10 shows the final location θ_{bump}mean for different values of I_0. An upper limit on I_0 that keeps the bump in a linear regime would be $I_0 \leq 0.47$ nA.

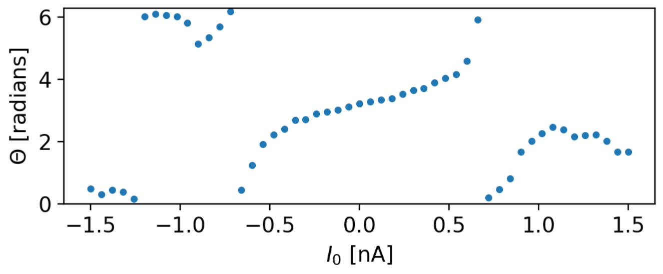


Fig. 10: Final location of θ_{bump}mean for different values of I_0

Ex.2.5: This system of coupled bump attractors can be viewed as a mechanism for integration of input due to its ability to represent and maintain a stable state that integrates information over time. Each population of neurons encodes a bump that represents a specific value or variable, and the connectivity between the populations enables the integration of these representations. External inputs can perturb the bumps, causing them to shift left or right, effectively updating the integrated variable. The mutual inhibition and excitation between the populations ensure that the system maintains equilibrium and integrates the input information, making it suitable for tasks requiring continuous integration and updating of variables.

Ex.3.1: The function creates a smooth random 2D trajectory with constant speed by incrementally updating the position and heading direction at each time step. The `step_size` ensures a uniform distance per step, while the `max_angle` parameter limits changes in the heading direction to avoid abrupt turns, ensuring smoothness. Starting from an initial position and heading, the function updates the position using trigonometric functions based on the current heading, which is gradually adjusted with small random perturbations.

Ex.3.2: Fig. 11 shows the results of simulating the head direction cells θ^H_{bump} againts the current head direction θ_H. The plot shows a strong correlation. This close match between the two, indicates that the head direction cells accurately track the actual head direction, demonstrating the effectiveness of the bump attractor network in integrating and representing head direction. Any significant mismatches could suggest that the turning speed in the trajectory generation was too fast, causing the neural network to lag.

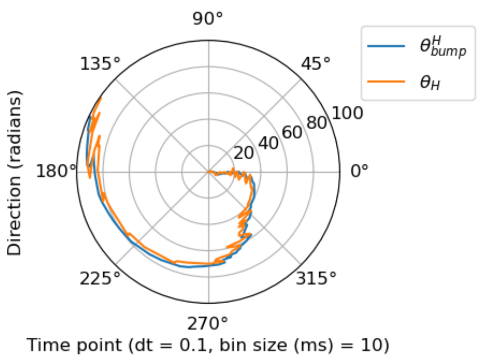


Fig. 11: θ^H_{bump} againts the current head direction θ_H.

Ex.3.3: Fig. 12 shows the values of J_{head} againts the maximum current of the head population to the other two couple attractors. Values of J_{head} close to 0.9 pC give rise to an input around 0.47 nA which as we saw in Fig. 10 puts the system in a linear regime.

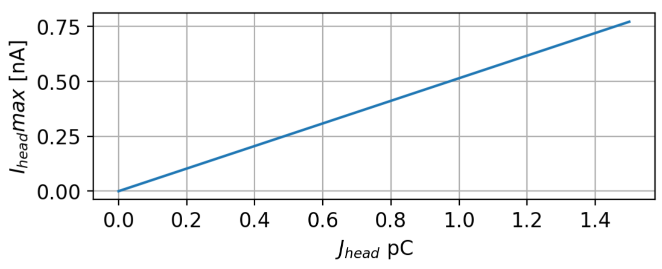


Fig. 12: θ_{bump}mean againts different values of J_{head}

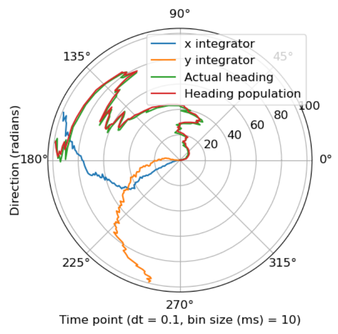
Ex.3.4: Equation 2 shows the equations for the input that is received by each of the five populations.

$$
\begin{aligned}
I_{head,i} &= I_0 cos(x_i^H - \theta_H(t)) \\
I_{left,i}^x &= \frac{J}{N}(\sum_{j=1}^{N} cos(x_i^L + \theta - x_j^L) S_j(t) \\
&+ \sum_{j=1}^{N} cos(x_i^L + \theta - x_j^R) S_j(t)) + \frac{J_{head}}{N}(\sum_{j=1}^{N} -cos(x_j^H) S_j(t)) \\
I_{right,i}^x &= \frac{J}{N}(\sum_{j=1}^{N} cos(x_i^R - \theta - x_j^L) S_j(t) \\
&+ \sum_{j=1}^{N} cos(x_i^R - \theta - x_j^R) S_j(t)) + \frac{J_{head}}{N}(\sum_{j=1}^{N} cos(x_j^H) S_j(t)) \\
I_{left,i}^y &= \frac{J}{N}(\sum_{j=1}^{N} cos(x_i^L + \theta - x_j^L) S_j(t) \\
&+ \sum_{j=1}^{N} cos(x_i^L + \theta - x_j^R) S_j(t)) + \frac{J_{head}}{N}(\sum_{j=1}^{N} -sin(x_j^H) S_j(t)) \\
I_{right,i}^y &= \frac{J}{N}(\sum_{j=1}^{N} cos(x_i^R - \theta - x_j^L) S_j(t) \\
&+ \sum_{j=1}^{N} cos(x_i^R - \theta - x_j^R) S_j(t)) + \frac{J_{head}}{N}(\sum_{j=1}^{N} sin(x_j^H) S_j(t))
\end{aligned} \quad (2)
$$

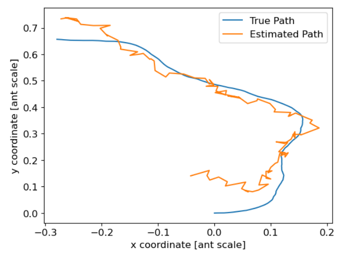
Ex.3.5: Fig. 13 shows the simulation of the network with $N = 300$ number of neurons and for $T = 1000$ miliseconds for the whole trajectory. The decoded trajectory (using a linear fit on the portion of data without the initial transient activity) is shown on top of the original trajectory. The imperfections in the decoded trajectory could be attributed to drifts, noise, and inaccuricies in the integration at are inherent to the simulation. The following parameters could be change to make the network perform betttter: 1) Decreasing the maximum angle change per step can reduce the likelihood of abrupt turns. 2) Increasing the number of neurons in the network can enhance the resolution and precision of the bump attractor. 3) Reducing noise in the system, either by modifying the spike generation mechanism or by using noise suppression techniques, can lead to a clearer signal and better decoding performance. 4) Reducing the step size can ensure that the head direction changes more smoothly and slowly, allowing the neural network to better track these changes and integrate the trajectory more accurately.

Ex.3.6: While the simplified network that we are using aids in computational modeling, real insect neural circuits might involve more complex interactions and dynamics. Moreover, we are using a model with fixed parameters but insects may have adaptive mechanisms to fine-tune these parameters based on sensory feedback and environmental conditions. The model spatially limited as well, as it wraps around when integrating more than π, while the "physical ant" does not.

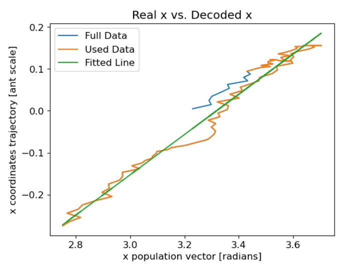
Ex.3.7: Fig. 14 shows the performance of the network with $N = 2000$ number of neurons and for $T = 4000$ miliseconds. The estimated trajectory is now more accurate and smoother.



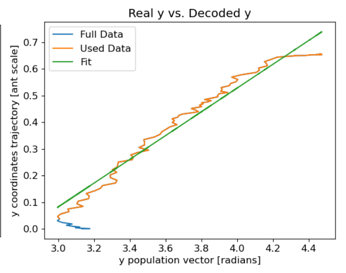
(a) Bump location for the x integrator, y integrator, and heading cell populations along with the actual heading direction.



(b) The estimated trajectory from the populations shown on top of thehground truth trajectory.

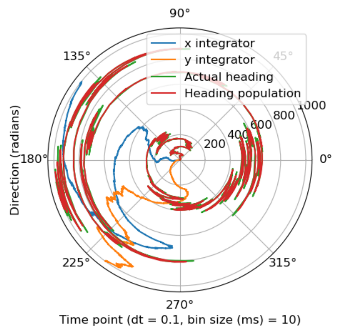


(c) The bump location for the x integrator population fitted againts the x position of the real trajectory.

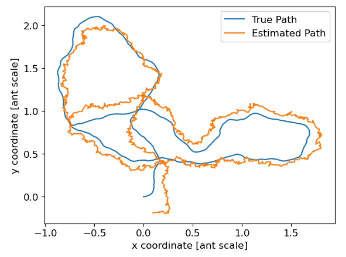


(d) The bump location for the y integrator population fitted againts the y position of the real trajectory.

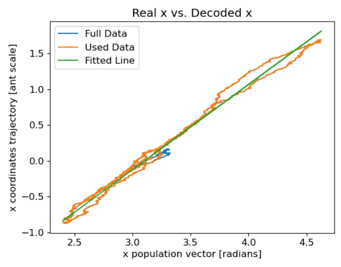
Fig. 13: Decoded activity of the two position integrators using a linear fit shown on top of the original trajectory.



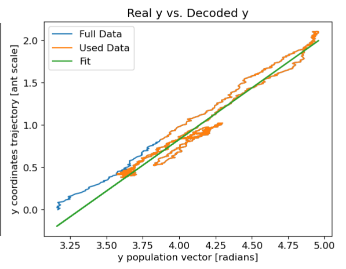
(a) Bump location for the x integrator, y integrator, and heading cell populations along with the actual heading direction.



(b) The estimated trajectory from the populations shown on top of thehground truth trajectory.



(c) The bump location for the x integrator population fitted againts the x position of the real trajectory.



(d) The bump location for the y integrator population fitted againts the y position of the real trajectory.

Fig. 14: Decoded activity of the two position integrators using a linear fit shown on top of the original trajectory.